Quantifying Dislocation Microstructures

Der Fakultät für Maschinenbau, Verfahrens- und Energietechnik
der Technischen Universität Bergakademie Freiberg

genehmigte

Dissertation

zur Erlangung des akademischen Grades

Doktor-Ingenieur
(Dr.-Ing.)

vorgelegt von M.Sc. Dominik Steinberger
geboren am 26. Oktober 1989 in Nürnberg

Gutachter:

- Prof. Dr. Stefan Sandfeld, Freiberg
- Prof. Dr. Alexander Hartmaier, Bochum

Tag der Verleihung: 20. Juli 2020

Versicherung

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Von folgende Personen habe ich Unterstützung bei dem Aufsetzen von Simulationen bzw. Unterstützung in Form von Durchführung von Experimenten bekommen:

- Daniel Kiener, Peter J. Imrich und Inas Issa haben die Experimente in chapter 4 durchgeführt und die Kraft-Verschiebungs-Daten sowie das Bildmaterial der Experimente zur Verfügung gestellt.

- Markus Stricker hat die diskreten Versetzungssimulationen, die in chapter 5 von mir analysiert werden, durchgeführt und die Rohdaten zur Verfügung gestellt.

- Hengxu Song hat mir bei der Erstellung der Parameterdatei für die diskreten Versetzungssimulationen in section 6.2 geholfen.

- Shyamal Roy hat mir die Eingangsdaten für die diskreten Versetzungssimulationen in section 6.3 zur Verfügung gestellt, sowie bei der Erstellung der Parameterdatei geholfen.

Weitere Personen waren an der Abfassung der vorliegenden Arbeit nicht beteiligt. Die Hilfe eines Promotionsberaters habe ich nicht in Anspruch genommen. Weitere Personen haben von mir keine geldwerten Leistungen für Arbeiten erhalten, die nicht als solche kenntlich gemacht worden sind.

Die Arbeit wurde bisher weder im Inland noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt.

Contents

List of tables

List of figures

Acknowledgments

This thesis is part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 759419).

I would first like to thank my supervisor, Prof. Dr. Stefan Sandfeld, for allowing me to pursue the research presented in this work and supporting me with invaluable advice whenever I had questions or doubts.

Next, I would like to thank Prof. Dr. Alexander Hartmaier, for having taken the time to closely examine this work and provide constructive feedback.

I would like to acknowledge my colleagues from the Institute of Materials Simulation in Erlangen, especially my former office mates Mehran Monavari and Stefan Liebenstein. Without you, the first half of my time as PhD student would not have been the same pleasure it was. I also would like to acknowledge my colleagues from the Institute of Mechanics and Fluid Dynamics in Freiberg, with a special shout-out to Rachel Strobl, Hengxu Song and Michael Budnitzki. Your support for this work is immeasurable.

Last, but definitely I want to thank my family and friends for their unwavering support. From the very get-go, my parents and sister kept my back clear so that I was able to pursue my interests that ultimately resulted in this thesis. A very special thank you to my partner Julia; if not for you the time spent researching would have been far less enjoyable.

Abbreviations

Abbreviation	**Long form**
BCC	body-centered cubic
CDD	continuum dislocation dynamics
CV	coefficient of variation
D2C	discrete-to-continuous
DDD	discrete dislocation dynamics
FCC	face-centered cubic
FEM	finite element method
FIB	focused ion beam
FR source	Frank–Read source
GBM	gradient boosting machine
GNB	Gaussian naïve Bayes
GND	geometrically necessary dislocation
HGBM	histogram-based gradient boosting machine
ID	identifier
*k*CV	*k*-fold cross-validation
LOOCV	leave-one-out cross-validation
MAD	mean absolute deviation
MAE	mean absolute error
MSE	mean squared error
PDE	partial differential equation

Abbreviation	**Long form**
RMSD	root-mean-square deviation
RMSE	root-mean-square error
SGD	stochastic gradient descent
S*k*CV	stratified *k*-fold cross-validation
SSD	statistically stored dislocation
STEM	scanning transmission electron microscopy
TEM	transmission electron microscopy

Notation

Notation	Description
$\tilde{\bullet}$	an approximated value
$\binom{\bullet}{\circ}$	binomial coefficient
$\boldsymbol{\bullet} \times \boldsymbol{\circ}$	cross product
$\langle h\,k\,l \rangle$	a family of crystallographic directions
$\boldsymbol{\bullet} \cdot \boldsymbol{\circ}$	dot product
$\bullet!$	factorial
$\check{\bullet}$	a given value
$\boldsymbol{\nabla} \boldsymbol{\bullet}$	gradient
$\max \bullet, \circ$	maximum
$\breve{\bullet}$	a predicted value
$\bar{\bullet}$	a prescribed value
$\circ \cap \bullet$	the intersection
$\circ \in \bullet$	the membership
$\{\bullet \mid \circ(\bullet)\}$	a set that is defined by a predicate
$\hat{\boldsymbol{\bullet}}$	a vector whose Euclidean norm is 1
$\boldsymbol{\bullet}$	a vector
$\lVert\bullet\rVert$	Euclidean norm
$\lvert\bullet\rvert$	absolute value
$\{\bullet_1, \bullet_2, \dots, \bullet_n\}$	a set that is defined by enumerating its elements
$\arg\max_{\circ} \bullet(\circ)$	arguments of the maximum
$\textsc{as}(\bullet, \circ)$	accuracy score
$\mathrm{Bias}\,[\bullet]$	bias
$\textsc{cv}\bullet$	coefficient of variation
$\operatorname{div} \boldsymbol{\bullet}$	divergence
$\textsc{e}(\bullet, \circ)$	error

Notation	**Description**
$\mathrm{E}\left[\bullet\right]$	expected value
$\langle\bullet\rangle$	the arithmetic mean of a sample
$[h\,k\,l]$	a crystallographic direction
$(h\,k\,l)$	a crystallographic plane
$\{h\,k\,l\}$	a family of crystallographic planes
$\bullet^{\otimes n}$	tensor power
$\bullet : \circ$	double dot product
$\bullet \otimes \circ$	tensor product
$\mathrm{MAD}\,\bullet$	mean absolute deviation
$\mathrm{MAE}\,\bullet$	mean absolute error
$\mathrm{MSE}\,\bullet$	mean squared error
$\bullet$	a tensor
$\mathrm{D}^{n}_{\circ}\,\bullet$	the derivative
$P(\bullet \mid \circ)$	conditional probability
$P(\bullet)$	probability
$\mathrm{RMSD}\,\bullet$	root-mean-square deviation
$\mathrm{RMSE}\,\bullet$	root-mean-square error
$\nabla_{\mathrm{S}}\,\bullet$	symmetric gradient
$\operatorname{sign}\bullet$	sign function
$\bullet^{T}$	transpose
$\operatorname{tr}\bullet$	the trace
$\mathrm{Var}\left[\bullet\right]$	variance

Symbols

Symbol	Description
a	regularization length of the Cai regularization function
A	area
$\boldsymbol{A}$, A_i	area vector
$\boldsymbol{\alpha}$, α_{ij}	Kröner–Nye tensor
$\boldsymbol{b}$, b_i	Burgers vector
$\mathscr{B}$	body
$\boldsymbol{B}$, B_i	body force
$\mathscr{C}$	curve
$\boldsymbol{C}$, C_{ijkl}	stiffness tensor
$\boldsymbol{d}$, d_i	slip direction
d	distance
$\boldsymbol{d}$, d_i	projection direction
$\boldsymbol{E}$, E_{ij}	Lagrangian strain tensor
E^{el}	elastic energy
e	element number
$\boldsymbol{\varepsilon}^{\mathrm{dis}}$, $\varepsilon_{ij}^{\mathrm{dis}}$	dislocation eigenstrain
$\boldsymbol{\varepsilon}^{\mathrm{eig}}$, $\varepsilon_{ij}^{\mathrm{eig}}$	eigenstrain
ϵ	error term
ϵ	slack variable
$\boldsymbol{\varepsilon}$, ε_{ij}	strain
$\boldsymbol{\varepsilon}^{\mathrm{tot}}$, $\varepsilon_{ij}^{\mathrm{tot}}$	total strain
ER	error rate
η	learning rate
$\boldsymbol{F}^{\mathrm{PK}}$, F_i^{PK}	Peach–Köhler force

Symbol	Description
$\boldsymbol{F}^{\mathrm{PK,cl}}$, $F_i^{\mathrm{PK,cl}}$	climb component of the Peach–Köhler force
$\boldsymbol{F}^{\mathrm{PK,gl}}$, $F_i^{\mathrm{PK,gl}}$	glide component of the Peach–Köhler force
$\boldsymbol{F}^{\mathrm{dg}}$, F_{ij}^{dg}	deformation gradient
$\boldsymbol{f}$, f_i	finite element right hand side
$\boldsymbol{F}$, F_i	force
f	a function
$\boldsymbol{g}$, g_i	diffraction vector
G	shear modulus
Γ	boundary
$\boldsymbol{I}$, I_{ij}	identity tensor
I	indicator variable
K	regularization function proposed by Cai et al. [1]
k	curvature
$\boldsymbol{K}$, K_{ij}	finite element matrix
L	gather operator
l	length
L	loss function
L	unit length
m	Schmid factor
M	margin
μ	the arithmetic mean of a population
$\boldsymbol{n}$, n_i	slip plane
$\boldsymbol{n}$, n_i	normal vector
n_{gp}	number of Gauss points
n_{dm}	number of dislocations microstructures
n_{dis}	number of dislocations
n_{en}	number of element nodes
n_{el}	number of elements
$n_{\mathrm{dim}}^{\boldsymbol{X}}$	number of input dimensions
n_{it}	number of iterations
n_{feat}	number of features

Symbol	**Description**
n_{obs}	number of observations
$n_{\Omega_{\text{sub}}}$	number of subvolumes
n_{V}	the number of vertices
N	shape function
ν	Poisson's ratio
Ω	domain
PFI	permutation feature importance
φ	azimuthal angle
$\boldsymbol{\varphi}$, φ_i	deformation mapping
π	ratio of a circle's circumference to its diameter
$\boldsymbol{q}^{(n)}$, $q^{(n)}_{i_1 i_2 \dots i_n}$	dislocation curvature density
r	radius
R	regularization term
ρ	the dislocation density
$\boldsymbol{\rho}^{(n)}$, $\rho^{(n)}_{i_1 i_2 \dots i_n}$	dislocation density alignment tensor
$\boldsymbol{\rho}^{(m,n)}$, $\rho^{(m,n)}_{i_1 i_2 \dots i_{m+n}}$	dislocation feature density tensor
ρ	higher-dimensional dislocation density
ρ^{t}	the total dislocation density
s_N	the standard deviation of a sample
σ	the standard deviation of a population
$\boldsymbol{\sigma}$, σ_{ij}	stress
t	curve parameter
t	time
$\boldsymbol{t}$, t_i	traction
τ^{rss}	resolved shear stress
θ	polar angle
$\boldsymbol{u}$, u_i	displacement
$\boldsymbol{v}$, v_i	a vertex
V	volume

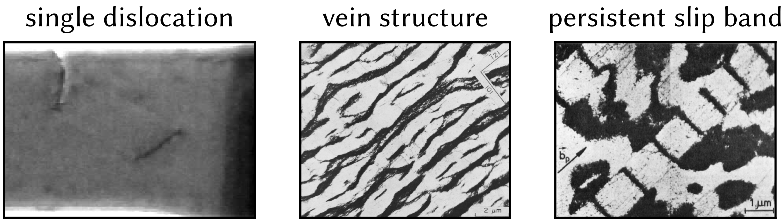
Symbol	**Description**
W	elastic energy density
w	weight
w	weight function
X	input of a machine learning model
$\boldsymbol{x}$, x_i	position
$\boldsymbol{x}$	x-axis
x	x-coordinate
$\boldsymbol{\xi}$, ξ_i	the sense of a dislocation
$\boldsymbol{\xi}$, ξ_i	tangent vector
Y	output of a machine learning model
$\boldsymbol{y}$	y-axis
y	y-coordinate
$\boldsymbol{z}$	z-axis
z	z-coordinate

1. Introduction

The late 5th millennium BCE marks the beginning of the Copper Age, and with it the advent of metalworking. Since then, humankind has exploited the malleability and ductility of metals to manufacture a variety or products and tools, ranging from delicate jewelry to massive crankshafts weighing several tens of metric tons. It is difficult to imagine modern society without the advances brought forth over the ages by ever improving tools made from metals and formed by forging, rolling, bending, drilling, milling, etc.

But it was not until 1923, when Taylor and Elam [2] studied the plastic deformation behavior of aluminum single crystals under tension via X-ray diffraction, that the systematic study of the underlying mechanisms of metal plasticity started to succeed. In 1934, Orowan [3, 4], Polanyi [5] and Taylor [6] independently proposed a line-like crystal defect as underlying cause of the plasticity of metals; the dislocation. First observations of dislocations soon followed in the early 1950s as microscopy improved significantly [7, 8].

Examples of dislocation microstructures that are observed in experiments are shown in Figure 1.1, with the dislocation microstructure being the dark portions of the images. Modern transmission electron microscopy imaging allows visualization of single dislocations, as seen on the left-hand side. In the center, we see a dislocation microstructure called vein structure that forms during fatigue experiments, i.e., experiments during which cyclic loading is applied. The veins comprise many parallel dislocations that form a bundle.



single dislocation

vein structure

persistent slip band

Figure 1.1.: Dislocation microstructures observed in experiments. A single dislocation in a micro-cantilever beam provided by D. Kiener, a vein structure observed during fatigue experiments [9], and a persistent slip band which is also observed during fatigue experiments [10].

Eventually, this structure breaks down locally and forms the ladderlike dislocation microstructure that is commonly referred to as persistent slip bands [11], which is depicted on the right. The sites at which these persistent slip bands intersect with specimen surfaces are preferential sites for crack initiation [12], and it is these cracks that commonly lead to failure of the material during experiments. As this failure mode is very prevalent in situations where metals are put under cyclic load, e.g., cars, airplanes or trains, it seems obvious that we would want to prevent the persistent slip band structure from forming in the first place. Unfortunately, we do not yet know how it forms exactly due to the collective behavior of dislocations being a very complex phenomenon.

Over the last few years, experimental methods improved to the point where they allow us to follow the motion of single dislocations in situ via transmission electron microscopy [13]. Even three-dimensional imaging is now possible [14, 15]. Due to advancements in both algorithms and computational power, numerical methods are able to simulate the evolution of dislocation microstructures at a length scale of several tens of micrometers [16] and provide more detailed insight into the underlying mechanisms of dislocation-based plasticity. Direct comparisons between experiments and simulations are potentially possible.

However, directly comparing two dislocation microstructures is difficult. How would we quantify the similarity between a vein structure and a persistent slip band? So far, dislocation microstructures are typically characterized via their density, i.e., the length of all dislocations divided by the specimen volume. But this description not only neglects physical properties of dislocations that are important for their characterization, it also does not take the spatial configuration of the microstructure into account. There is a myriad of ways the same total length of dislocations inside a volume can be oriented and concentrated, resulting in a vast amount of degrees of freedom with respect to their spatial configuration. So-called continuum theories of dislocations use specialized field descriptions of dislocation microstructures to model their behavior [17, 18, 19, 20, 21, 22] with a degree of accuracy that depends on the respective formulation [23]. If these field descriptions allow such modeling, they likely consider the physical properties of dislocations that are required to characterize them. Hence, they may be also be used as a means of describing dislocation microstructures in a way that makes them comparable in more detail than just their line length in a whole volume.

But even though there is a rapidly growing body of dislocation microstructure data, the inherent complexity of this data with its immense number of degrees of freedom requires the development of suitable algorithms and methods to enable their analysis. The recent resurgence of machine learning might aid in this endeavour. Other fields within materials science have seen a lot of success when applying machine learning on complex problems, e.g., material-structure-property relationships [24], microstructure characterization [25, 26], band gap prediction [27], prediction of the crystal structure [28], and prediction of stable compounds [29]. However, successful application of machine learning techniques requires both, a sufficient amount of data and a method of extracting features from the data such that they can be used as input for machine learning algorithms.

Due to the large number of degrees of freedom of dislocation microstructures, we likely require a very large amount of data if we want to exploit modern statistical and machine learning methods. Ideally, this data would be three-dimensional dislocation data, which is still difficult to obtain from experiments. But even if we had the necessary data, how would we extract features from dislocation microstructures that can be used for machine learning? Aforementioned continuum theory field descriptions might be suitable, and we investigate this in the present work.

The goal of this work is three-fold. First, we want to show how using computational analyses enable us to acquire three-dimensional dislocation data from two-dimensional image data obtained from experiments performed at the micrometer length scale. Second, we want to demonstrate how the description of dislocation microstructures via continuous fields allows us to quantify the similarity of different dislocation microstructures. Last, we want to study the potential of using a large amount of dislocation microstructure data within the context of machine learning.

Structure of the thesis

In chapter 2, we introduce the theoretical fundamentals that are the basis of this thesis. After a brief summary of important relationships within linear elasticity, we proceed to cover general dislocation theory. We outline the geometrical and physical properties of dislocations, how they behave in the presence of stresses and surfaces, and how they induce stresses within crystalline materials. Finally, we summarize common density description of dislocations which are used throughout this work to quantify dislocation microstructure features.

The methods used to perform analyses and simulations within this thesis are outlined in chapter 3. We first cover the finite element method as a means for numerically solving linear elastic problems and go into more detail on how to consider the contribution of dislocations within it. Next, we summarize the discrete dislocation dynamics method which we use to simulate the evolution of dislocations under different types of loading. We then introduce the discrete-to-continuous method, which formalizes how discrete dislocation data can be converted to continuum fields that enable quantitative comparisons of different dislocation microstructures. Subsequently, we summarize the basic concepts behind machine learning, how to assess the predictive power of a machine learning model and how to extract relevant information from them. In this context, we also briefly outline the specifics of using dislocation densities as input for machine learning models.

In chapter 4, we show how the combination of state-of-the-art in situ microscopy experiments and finite element method analyses enable the reconstruction of three-dimensional dislocation configurations from transmission electron microscopy images. We analyze a micro-beam cantilever experiment during which dislocations are emitted from a notch. By combining the information obtained from the experiment with insight from the computational analysis, we are able to fully characterize the dislocations and reconstruct their three-dimensional configuration. Based on this configuration, we are then able to infer likely intermediate states of the propagation of the dislocations in this experiment.

We study the impact of initial conditions for discrete dislocation dynamics on the dislocation microstructure that forms during simulations in chapter 5. From previous studies, it is known that the initial dislocation microstructure affects, e.g., the stress-strain response or the evolution of the total dislocation density within discrete dislocation dynamics simulations. Based on our approach we additionally are able to quantifying how different the obtained dislocation microstructures are and what the implications for simulations are.

Chapter 6 is dedicated to using continuous dislocation density fields as input for machine learning methods. To this end, we perform two studies during which we train machine learning models to predict the size of a particle in which a dislocation microstructure has formed. In one case, the dislocation microstructures are obtained by allowing random dislocation microstructures in particles of different size to relax. In the other case, the dislocation microstructures form during indentation of differently sized particles. We test a variety of parameter combinations and study their effect on the predictive power of the machine learning models. Additionally, we probe how these they use of the different dislocation microstructure information they are provided with. Combined, this information allows us to infer the features that set apart dislocation microstructures that emerged in different particle sizes.

In the last chapter, chapter 7, we frame the findings of our three studies in the context of databases of dislocation microstructures. We provide recommendations on how to improve the collection of dislocation microstructure data from experiments, and how to potentially use the data to gain more insight into underlying mechanisms of dislocation-based plasticity.

2. Theory

In this chapter, we outline the theoretical foundation necessary for this work. We start by introducing the relevant terms and quantities and introduce the concept and formulation of linear elasticity. For this, we follow the notation of Tadmor and Miller [30]. Subsequently, we lay out the definition of dislocations along with their most important properties. We then cover why dislocations move within a crystal and how they interact with other dislocations and free surfaces. Finally, we summarize commonly used continuum representations of dislocations that we will use for the quantification of dislocation microstructures.

2.1. Linear Elasticity

When analyzing the deformation of a body $\mathscr{B}$ under external load, the unloaded state may be taken as a reference configuration $\mathscr{B}_0$, to which the deformed state can be compared. The connection between the reference and the deformed state is the so-called deformation mapping $\boldsymbol{\varphi}$, which maps each position $\boldsymbol{x}_0$ of the reference configuration to a position $\boldsymbol{x}$ within the deformed configuration. One way to describe this deformation mapping is by means of a displacement field $\boldsymbol{u}(\boldsymbol{x}_0)$, by which a material point at position $\boldsymbol{x}$ is displaced from its initial position $\boldsymbol{x}_0$:

$$\boldsymbol{x} = \boldsymbol{\varphi}(\boldsymbol{x}_0) = \boldsymbol{x}_0 + \boldsymbol{u}(\boldsymbol{x}_0) \tag{2.1}$$

Subsequently, the deformation mapping can be used to define the deformation gradient

$$\boldsymbol{F}^{\mathrm{dg}} = \frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{x}_0} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{x}_0} = \nabla_0\, \boldsymbol{x} = \boldsymbol{I} + \nabla_0\, \boldsymbol{u} \tag{2.2}$$

as a means of describing the local deformation in the neighborhood of a material point. The Lagrangian strain tensor

$$\boldsymbol{E} = \frac{1}{2}\left(\boldsymbol{F}^{\mathrm{dg}\,T}\boldsymbol{F}^{\mathrm{dg}} - \boldsymbol{I}\right) = \frac{1}{2}\left(\nabla_0\, \boldsymbol{u} + (\nabla_0\, \boldsymbol{u})^T\right) + \frac{1}{2}(\nabla_0\, \boldsymbol{u})^T \nabla_0\, \boldsymbol{u}, \tag{2.3}$$

where $\bullet^T$ denotes the transpose, can then be defined to describe how the vicinity of a point “stretches”, i.e., without taking rigid body rotation into account.

When the deformation is small, and hence the displacement gradient is small, the quadratic term may be dropped, which leads to the linear strain tensor

$$\boldsymbol{\varepsilon} = \frac{1}{2}\left(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^{T}\right) = \nabla_{\mathrm{S}}\, \boldsymbol{u}, \tag{2.4}$$

where $\nabla_{\mathrm{S}} \bullet$ is defined as the symmetric gradient. Later on, the linearity of this strain measure is used to consider eigenstrains.

Ultimately, strain results in stress $\boldsymbol{\sigma}$ in a material. For a body to be in mechanical equilibrium, both the balance of linear momentum

$$\operatorname{div} \boldsymbol{\sigma} + \boldsymbol{B} = 0, \tag{2.5}$$

and the balance of angular momentum

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}^{T} \tag{2.6}$$

have to be fulfilled at every position within the body. The connection between strain and stress is governed by the generalized Hooke's law

$$\boldsymbol{\sigma} = \boldsymbol{C} : \boldsymbol{\varepsilon}, \tag{2.7}$$

with $\boldsymbol{C}$ being the so-called stiffness tensor, a rank four tensor with 81 components. Due to the symmetry of the strain tensor (2.4), the balance of angular momentum (2.6), and additional symmetries that are outlined in Tadmor and Miller [30], the number of independent components of the stiffness tensor reduces to 21. For convenience, equation (2.7) can be written in the so-called Voigt notation, i.e., a contracted matrix notation. While we do not use this notation in the present work, the components of $\boldsymbol{C}$ are commonly provided using their matrix indices, where the first and second index pair are represented by one index each. Their mapping is given as:

tensor indices	11	22	33	23, 32	13, 31	12, 21
matrix index	1	2	3	4	5	6

Material symmetries may further decrease the number of independent components. For cubic materials, the number of independent components is three. In matrix notation, these are $C_{11} = C_{22} = C_{33}$, $C_{12} = C_{23} = C_{13}$, and $C_{44} = C_{55} = C_{66}$. All other components are zero. In isotropic materials, one of these components is a function of the other two, resulting in two independent components. One way they may be provided are via the shear modulus G and the Poisson's ratio ν.

Further measures that are used within this work are the traction $\boldsymbol{t}$ and the elastic energy density W. Tractions are forces that act on a surface with normal vector $\boldsymbol{n}$. They are related to stresses via

$$\boldsymbol{t}(\boldsymbol{n}) = \boldsymbol{\sigma} \cdot \boldsymbol{n}. \tag{2.8}$$

Stress and strain are linked to the elastic energy density via

$$W = \frac{1}{2}\boldsymbol{\sigma} : \boldsymbol{\varepsilon}. \tag{2.9}$$

2.2. Dislocations

Dislocations are line defects that occur in crystals. They represent "the boundary of an area across which a relative slip displacement has occurred." [31] Their existence as crystalline defect was brought forward by Orowan [3], Orowan [4], Polanyi [5], and Taylor [6] as an explanation for the discrepancy between the theoretically predicted shear strength of a perfect crystal [32] and the experimentally observed ones. Instead of whole crystal halves shearing as a whole, dislocations move and expand or shrink the sheared area. This mode of deformation requires less load and therefore results in lower shear strength of crystalline materials. For this reason, dislocations are also referred to as the "carriers" of plastic deformation. And while the behavior of single dislocations is rather well understood, their interactions with each other can lead to complex behaviors. In the following, we outline the theoretical groundwork necessary to understand how the quantification of dislocation microstructures can be performed and applied within a data-driven approach.

2.2.1. Geometric properties

As the dislocation line is the boundary of an area, it can not end arbitrarily. It either forms a closed loop, or terminates at another defect as, e.g., a grain boundary, another dislocation, or at a free surface. The area enclosed by the dislocation is its dislocation slip area. The relative slip displacement across this area is called the Burgers vector $\boldsymbol{b}$ of the dislocation. To determine the Burgers vector, we may use the so-called Burgers circuit, which is depicted in Figure 2.1. For this, we compare a pristine crystal with one containing a dislocation parallel to the viewing direction, indicated by $\otimes$. First, we have to decide whether the sense $\boldsymbol{\xi}$ of the dislocation points into the paper or out of this. In this case, let us assume it points into the paper. We then form a right-handed circuit around $\boldsymbol{\xi}$ in the pristine crystal. Within Figure 2.1, it would

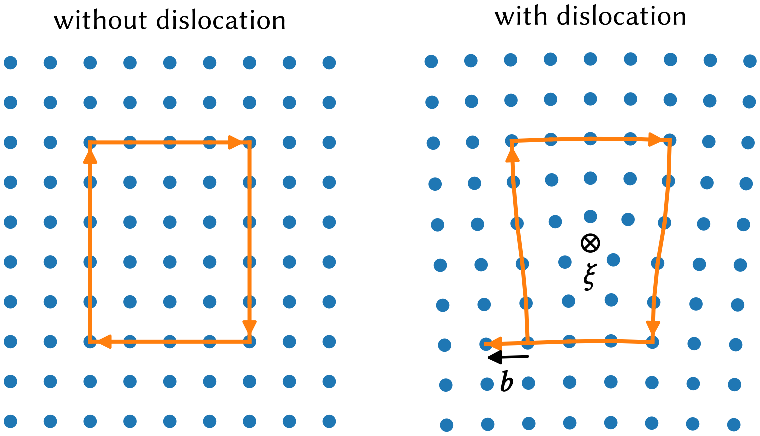


Figure 2.1.: Depiction of a Burgers circuit. A reference circuit starting at the lower left corner is performed in the pristine crystal (left). The same sequence – five atoms up, four to the right, five down, four to the left – is repeated around the dislocation (right). Subsequently, the Burgers vector may be computed as the difference between the final and starting points of the circuit.

be five atoms up, four to the right, five down, and four to the left. When we repeat this exact sequence for the configuration that contains the dislocation, we do not end up on the atom we started at. The vector connecting the starting atom with the finishing atom then is the Burgers vector $\boldsymbol{b}$. From the definition of the Burgers circuit follows, that the direction of the Burgers vector $\boldsymbol{b}$ of a dislocation is reversed when its sense $\boldsymbol{\xi}$ is reversed.

In the aforementioned example, the Burgers vector is perpendicular to the sense of the dislocation. This is not necessarily the case along a dislocation line, and the relative orientation of the Burgers vector to the sense locally defines the character of a dislocation. A dislocation is of edge character at position $\boldsymbol{x}$, when its Burgers vector is perpendicular to its sense at this position, i.e., $\boldsymbol{b} \perp \boldsymbol{\xi}(\boldsymbol{x})$. When they are parallel, i.e., $\boldsymbol{b} \parallel \boldsymbol{\xi}(\boldsymbol{x})$, the dislocation is of screw character at this position. In between those cases, the dislocation is of mixed character.

As the material deforms, dislocations propagate through the material. Once two dislocations get close, they may react depending on their Burgers vectors and senses. One possible reaction is the formation of a common dislocation line, also called a junction. It is a common boundary of two different dislocation slip areas with potentially different Burgers vectors. The points where these three dislocation lines meet are called dislocation nodes. When the senses of all n dislocations meeting at such a dislocation node are taken to point away

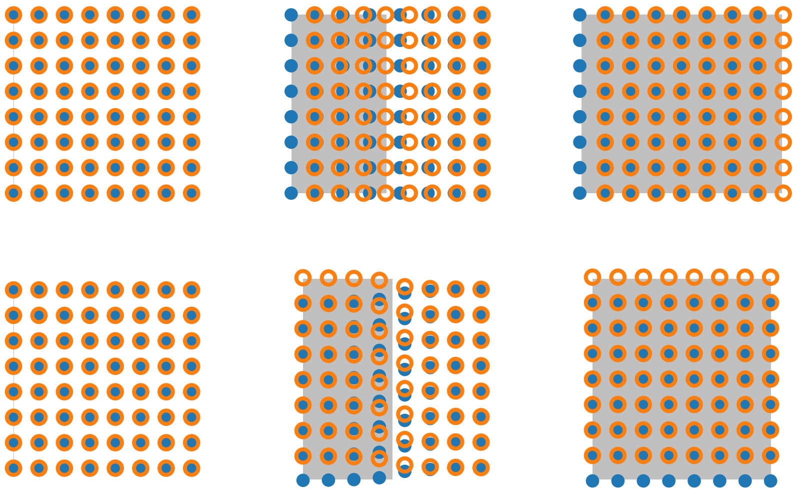


Figure 2.2.: Propagation of an edge and a screw dislocation through a crystal, shown in the top and bottom row, respectively. The dislocation slip area is shown as shaded area.

from it, then

$$\sum_{i=1}^{n} \boldsymbol{b}_i = \boldsymbol{0}. \tag{2.10}$$

As we will outline in the following, this might impede the motion of the dislocations involved in the junction.

2.2.2. Dislocation motion and driving forces

The motion of dislocations is important as it means that the accompanying deformation is propagating through the material, which is illustrated in Figure 2.2. A sufficient amount of dislocation motion can result in considerable plastic deformation of a material. In the following, we want to outline the fundamentals of how dislocations move and what the driving forces for dislocation motion are. In general, two different modes of motion can be distinguished for dislocations; *conservative* and *nonconservative* motion.

Conservative motion of a dislocation means that no matter has to be removed for the dislocation to move. This is the case if a dislocation with Burgers vector $\boldsymbol{b}$ and sense $\boldsymbol{\xi}$ moves within a plane whose normal vector is parallel to $\boldsymbol{b} \times \boldsymbol{\xi}$. Often, this is also referred to as glide or slip. Whenever a dislocation moves outside a aforementioned plane, matter has to be removed, e.g., by means of diffusion of interstitials or vacancies. This is commonly called climb motion.

Table 2.1.: Possible slip system families and number of slip systems within each slip system family for the FCC crystal structure.

slip system family	number of slip systems
$\{1\,1\,1\}\langle 1\,1\,0\rangle$	12
total	12

Table 2.2.: Possible slip system families and number of slip systems within each slip system family for the BCC crystal structure.

slip system family	number of slip systems
$\{1\,1\,0\}\langle 1\,1\,1\rangle$	12
$\{2\,1\,1\}\langle 1\,1\,1\rangle$	12
$\{3\,2\,1\}\langle 1\,1\,1\rangle$	24
total	48

Due to slip not requiring the removal of matter, it is in general easier for a dislocation to move via slip instead of climb. Note, that for dislocation segments of pure screw character $\boldsymbol{b} \times \boldsymbol{\xi} = \boldsymbol{0}$. A screw dislocation is therefore not limited to a slip plane, and could move in conservative fashion on all planes that are parallel to its Burgers vector $\boldsymbol{b}$. The change of slip plane it is moving on is called cross-slip. It provides an additional degree of freedom in the motion of dislocations.

The combination of slip plane and the direction of the Burgers vector – the so-called slip direction – form a slip system. In single crystals, certain slip system are more often observed to exhibit dislocation activity. They generally comprise close-packed slip directions combined with close-packed slip planes. The present work studies dislocation microstructures in face-centered cubic (FCC) and body-centered cubic (BCC) crystal structures. Their commonly observed slip systems are summarized in tables 2.1 and 2.2.

The driving force acting on dislocations on a specific slip system with slip plane $\boldsymbol{n}$ and slip direction $\boldsymbol{d}$ is the resolved shear stress

$$\tau^{\mathrm{rss}} = \boldsymbol{d} \cdot \boldsymbol{\sigma} \cdot \boldsymbol{n}. \tag{2.11}$$

It could also be described as the part of the stress acting on the slip plane in the slip direction. In simple tensile tests, we may compute of the resolved shear stress for all candidate slip systems. The slip system(s) exhibiting the most resolved shear stress typically predominates within this test. For a simple tensile load along a coordinate-axis the relationship between the non-zero stress component σ_{ii} and the resolved shear stress of a slip system can also be expressed via

$$\tau^{\mathrm{rss}} = m\sigma_{ii}, \tag{2.12}$$

where $m = d_i n_i$ is the so-called Schmid factor. Due to the slip plane being perpendicular to the slip plane follows that $0 \leq m \leq 0.5$.

A more general concept of describing the driving force of dislocations is the Peach–Köhler force

$$\frac{\boldsymbol{F}^{\mathrm{PK}}}{L} = (\boldsymbol{b} \cdot \boldsymbol{\sigma}) \times \hat{\boldsymbol{\xi}}, \tag{2.13}$$

which describes the force per unit length acting on a dislocation segment with Burgers vector $\boldsymbol{b}$ and sense $\boldsymbol{\xi}$ due to a stress $\boldsymbol{\sigma}$. The Peach–Köhler force is not limited in its direction, i.e., it is not necessarily parallel to the slip plane and may contain an out-of-plane component. But as we outlined before, dislocation motion is typically categorized in glide and climb. As these modes of motion exhibit different barriers for a dislocation to actually move, it is useful to distinguish between the glide component of the Peach–Köhler force

$$\frac{\boldsymbol{F}^{\mathrm{PK,gl}}}{L} = \frac{\left((\boldsymbol{b} \cdot \boldsymbol{\sigma}) \times \hat{\boldsymbol{\xi}}\right) \cdot \left(\hat{\boldsymbol{\xi}} \times (\boldsymbol{b} \times \hat{\boldsymbol{\xi}})\right)}{\|\boldsymbol{b} \times \hat{\boldsymbol{\xi}}\|} \tag{2.14}$$

and the climb component of the Peach–Köhler force

$$\frac{\boldsymbol{F}^{\mathrm{PK,cl}}}{L} = \frac{\left((\boldsymbol{b} \cdot \boldsymbol{\sigma}) \times \hat{\boldsymbol{\xi}}\right) \cdot (\boldsymbol{b} \times \hat{\boldsymbol{\xi}})}{\|\boldsymbol{b} \times \hat{\boldsymbol{\xi}}\|}. \tag{2.15}$$

The origin of stresses that affect dislocations is not limited to external loading. One of the most important sources of stress acting on a dislocation are actually other dislocations and the dislocation itself.

2.2.3. Dislocation induced stress fields

For the general case of a possibly curved dislocation in a finite body and with anisotropic elasticity, no analytic expression exists. However, we can use the ones formulated for particular edge cases to gain insight into how dislocations might affect each other. First, we cover the case of infinite, straight dislocation in an isotropic, infinite medium.

2.2.3.1. Infinite, straight dislocations in an isotropic, infinite medium

We provide the expressions for the two cases of a pure screw and a pure edge dislocation. These expressions follow from treating dislocations within linear elasticity. Thus, expressions for mixed dislocation character may be composed by superimposing these two solutions using prefactors that account for the ratio of the character.

The stress field associated with a straight screw dislocation in an infinite, continuous and isotropic medium is given by [31]

$$\sigma_{xx} = 0, \tag{2.16a}$$
$$\sigma_{yy} = 0, \tag{2.16b}$$
$$\sigma_{zz} = 0, \tag{2.16c}$$
$$\sigma_{xy} = 0, \tag{2.16d}$$
$$\sigma_{yz} = \frac{G\|\boldsymbol{b}\|}{2\pi}\frac{x}{x^2+y^2}, \text{ and} \tag{2.16e}$$
$$\sigma_{zx} = -\frac{G\|\boldsymbol{b}\|}{2\pi}\frac{y}{x^2+y^2}, \tag{2.16f}$$

with the non-zero components being shown on the left-hand side of Figure 2.3.

The stress field associated with a straight edge dislocation in an infinite, continuous and isotropic medium is given by [31]

$$\sigma_{xx} = -\frac{G\|\boldsymbol{b}\|}{2\pi(1-\nu)}\frac{y\left(3x^2+y^2\right)}{\left(x^2+y^2\right)^2} \tag{2.17a}$$
$$\sigma_{yy} = \frac{G\|\boldsymbol{b}\|}{2\pi(1-\nu)}\frac{y\left(x^2-y^2\right)}{\left(x^2+y^2\right)^2} \tag{2.17b}$$
$$\sigma_{zz} = -\frac{G\|\boldsymbol{b}\|\nu}{\pi(1-\nu)}\frac{y}{x^2+y^2} \tag{2.17c}$$
$$\sigma_{xy} = \frac{G\|\boldsymbol{b}\|}{2\pi(1-\nu)}\frac{x\left(x^2-y^2\right)}{\left(x^2+y^2\right)^2} \tag{2.17d}$$
$$\sigma_{yz} = 0 \tag{2.17e}$$
$$\sigma_{zx} = 0 \tag{2.17f}$$

with the non-zero components being shown on the right-hand side in Figure 2.3.

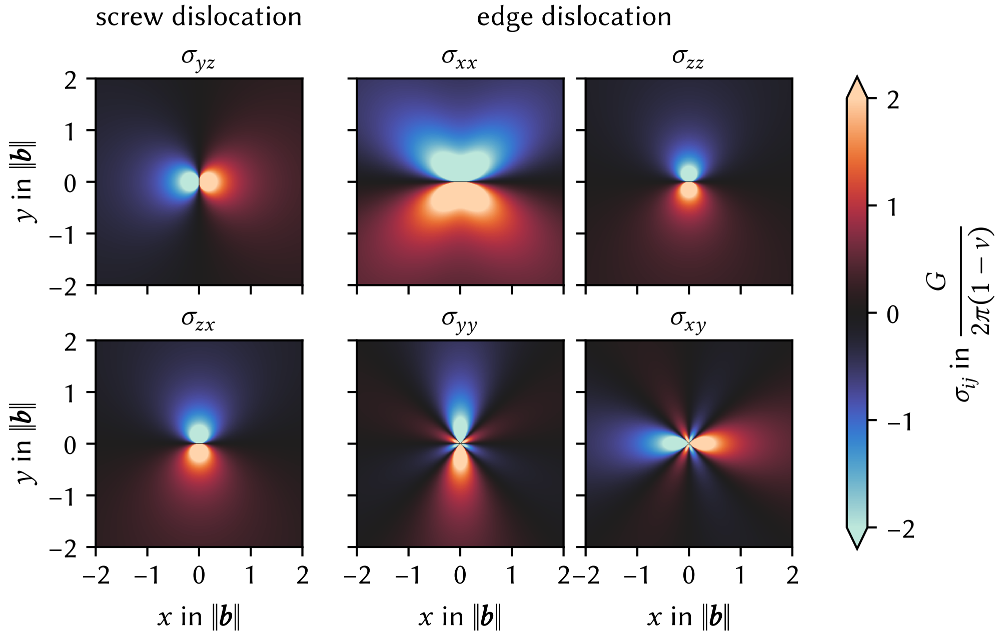


Figure 2.3.: Non-zero components of the stress field $\boldsymbol{\sigma}$ of infinite straight dislocations in an infinite medium.

These expressions are derived by assuming that the dislocation – and therefore the boundary of the dislocation slip area – is infinitely sharp. As such, it does not take the actual crystal lattice into account, and from atomistic simulations we know that the actual dislocation core has finite spread. Within the former treatment, the stress fields diverge as the distance to the dislocation goes towards zero. This is not observed in the more accurate atomic calculations. A side-effect of such a stress singularity would be that a curved dislocation exerts infinite stress onto itself [1].

Cai et al. [1] propose to take this into account by describing the finite dislocation core by "spreading the Burgers vector isotropically about every point on the dislocation line using a spreading function." The resulting non-singular expressions for the stress fields of an infinite, straight screw dislocation in an isotropic, infinite medium are given by

$$\sigma_{xx}^{\text{ns}} = 0, \tag{2.18a}$$

$$\sigma_{yy}^{\text{ns}} = 0, \tag{2.18b}$$

$$\sigma_{zz}^{\text{ns}} = 0, \tag{2.18c}$$

$$\sigma_{xy}^{\text{ns}} = 0, \tag{2.18d}$$

$$\sigma_{yz}^{\text{ns}} = \frac{G\|\boldsymbol{b}\|}{2\pi}\frac{x}{x^2+y^2+a^2}\left(1+\frac{a^2}{x^2+y^2+a^2}\right), \text{ and} \tag{2.18e}$$

$$\sigma_{zx}^{\text{ns}} = -\frac{G\|\boldsymbol{b}\|}{2\pi}\frac{y}{x^2+y^2+a^2}\left(1+\frac{a^2}{x^2+y^2+a^2}\right), \tag{2.18f}$$

and for an edge dislocation by

$$\sigma_{xx}^{\text{ns}} = -\frac{G\|\boldsymbol{b}\|}{2\pi(1-\nu)}\frac{y}{x^2+y^2+a^2}\left(1+\frac{2(x^2+a^2)}{x^2+y^2+a^2}\right) \tag{2.19a}$$

$$\sigma_{yy}^{\text{ns}} = \frac{G\|\boldsymbol{b}\|}{2\pi(1-\nu)}\frac{y}{x^2+y^2+a^2}\left(1-\frac{2(y^2+a^2)}{x^2+y^2+a^2}\right) \tag{2.19b}$$

$$\sigma_{zz}^{\text{ns}} = -\frac{G\|\boldsymbol{b}\|\nu}{\pi(1-\nu)}\frac{y}{x^2+y^2+a^2}\left(1+\frac{a^2}{x^2+y^2+a^2}\right) \tag{2.19c}$$

$$\sigma_{xy}^{\text{ns}} = \frac{G\|\boldsymbol{b}\|}{2\pi(1-\nu)}\frac{x}{x^2+y^2+a^2}\left(1-\frac{2y^2}{x^2+y^2+a^2}\right) \tag{2.19d}$$

$$\sigma_{yz}^{\text{ns}} = 0 \tag{2.19e}$$

$$\sigma_{zx}^{\text{ns}} = 0. \tag{2.19f}$$

A comparison of the linear elastic solution and the non-singular expression is shown in Figure 2.4.

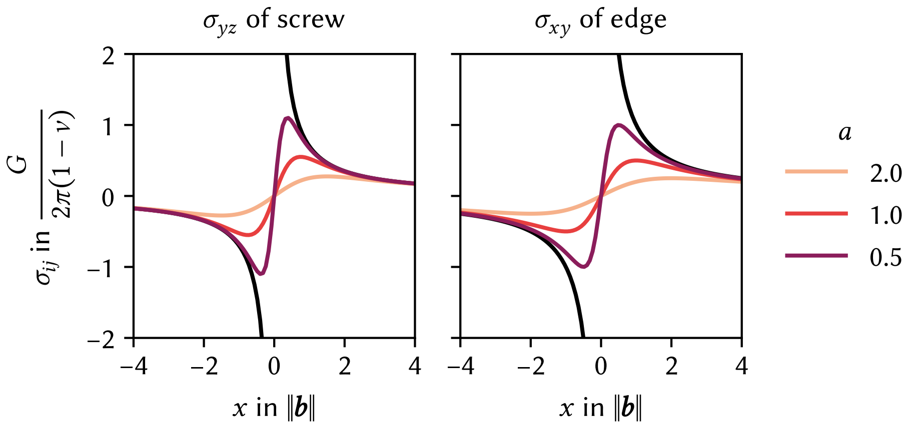


Figure 2.4.: Comparison of the linear elastic solution of infinite, straight dislocations in an infinite medium (black), and the corresponding non-singular expression by Cai et al. [1] for $y = 0$, i.e., within the slip plane. Colors denote different values of the regularization length a.

However, we are interested in dislocation configurations that comprise curved dislocations as well. One strategy is to approximate the dislocation line by straight line segments and use analytical stress field expression available for them. An easy to implement expression in tensor and vector notation for this case was derived by Devincre [33]. In a similar fashion, Cai et al. [1] also provides an expression for the non-singular solution in tensor and vector notation. Still, the limitations of only being valid for an isotropic elastic material in an infinite medium still applies. We eventually want to determine the stress field in finite bodies and take elastic anisotropy into account. To this end, we may utilize an eigenstrain ansatz.

2.2.3.2. Eigenstrain-based computation of dislocation stress fields

The name eigenstrain was first used by Mura [34] to describe “nonelastic strains as thermal expansion, phase transformation, initial strains, plastic strains, and misfit strains.” In his work, he proposes that – within linear elasticity – a total strain $\boldsymbol{\varepsilon}^{\text{tot}}$ may be introduced that is the sum of the “elastic” strain $\boldsymbol{\varepsilon}$ and the eigenstrain $\boldsymbol{\varepsilon}^{\text{eig}}$

$$\boldsymbol{\varepsilon}^{\text{tot}} = \boldsymbol{\varepsilon} + \boldsymbol{\varepsilon}^{\text{eig}}. \tag{2.20}$$

Using this relation, the generalized Hooke’s law (2.7) may be written as

$$\boldsymbol{\sigma} = \boldsymbol{C} : (\boldsymbol{\varepsilon}^{\text{tot}} - \boldsymbol{\varepsilon}^{\text{eig}}). \tag{2.21}$$

The eigenstrain-contribution $\mathrm{d}\boldsymbol{\varepsilon}^{\mathrm{dis}}$ of a dislocation at each point of the dislocation slip area is then given by

$$\mathrm{d}\boldsymbol{\varepsilon}^{\mathrm{dis}} = \frac{1}{2}(\boldsymbol{b} \otimes \mathrm{d}\boldsymbol{A} + \mathrm{d}\boldsymbol{A} \otimes \boldsymbol{b}), \tag{2.22}$$

where $\mathrm{d}\boldsymbol{A}$ is a vector parallel to the normal vector of the slip plane, and whose magnitude is the area associated with the point.

In this way, dislocations and their resulting stress fields can be considered within linear elastic problems. Dislocations are taken into account via their eigenstrain contributions. The problem is solved for the total strain, either via the displacements or directly. Subsequently, the stresses $\boldsymbol{\sigma}$ are then computed via (2.21).

In the present work, we use this eigenstrain-based method for the computation of the dislocation stress fields within the finite element method, which is outlined in section 3.1.

2.2.4. Dislocation interaction with free surfaces

Once we consider finite bodies, we have to take into account how dislocations interact with free surfaces. Free surfaces imply that the traction acting perpendicular to that surface must be zero. These boundary conditions also hold true in the presence of dislocations, and finally result in forces acting upon them.

To illustrate the basic interaction, we make use of the simple case of an infinite straight dislocation parallel to a single free surface first. Consider an infinite, straight dislocation in an infinite half-space, whose boundary corresponds to a free surface that is both parallel to the dislocation and perpendicular to its slip plane. The distance between the dislocation and the free surface is denoted by d. A screw dislocation would then be attracted towards the surface by a force per unit length

$$\frac{\boldsymbol{F}}{L} = \frac{G\|\boldsymbol{b}\|^2}{4\pi d}, \tag{2.23}$$

an edge dislocation by

$$\frac{\boldsymbol{F}}{L} = \frac{G\|\boldsymbol{b}\|^2}{4\pi(1-\nu)d}. \tag{2.24}$$

We can see that the attractive force exerted on a dislocation are stronger, the closer the dislocation is to the surface.

If a dislocation is inclined to the surface, the force acting on it additionally depends on the angle between the surface and the dislocation. A stable configuration in this case would be if the dislocation were perpendicular to the free surface. However, in more realistic configurations, dislocations are not necessarily able to move freely to either leave the body through the free surface or orient itself perpendicular to the surface entirely. But based on the attractive forces, we can expect fewer dislocations in the vicinity of a free surface, and that these dislocations show a preferred orientation that is perpendicular to the surface.

2.2.5. Continuum representations of dislocations

Dislocations are discrete objects. However, in some circumstances we benefit from deriving density-like quantities for dislocations. One example would be to compare the number of dislocations that are visible on transmission electron microscopy (TEM) images. Another one would be simulation methods that evolve dislocation microstructures in time via density like quantities instead of as discrete objects, which can lead to higher computational efficiency. The latter require continuum descriptions of dislocations which faithfully represent the underlying physics.

2.2.5.1. Total dislocation density

All these theories use density-like quantities to represent dislocations, the simplest one being the total dislocation density. The total dislocation density within a domain Ω is defined as

$$\rho^{\mathrm{t}} = \frac{1}{V_\Omega} \sum_{i=1}^{n_{\mathrm{dis}}} l_i^{\mathrm{dis}}, \tag{2.25}$$

where V_Ω is the volume of the domain that contains n_{dis} dislocations, each of which has a length l^{dis}. Note that we assume all dislocations to be contained in their entirety in the domain.

But as we have outlined in section 2.2.1, dislocations are defined by more than just their spatial configuration. These additional properties have to be taken into account for a continuum theory of dislocations that is supposed to model their behavior.

2.2.5.2. Kröner–Nye tensor

A density-like quantity that takes more than just the spatial configuration of the dislocation into account is the Kröner–Nye tensor [35, 36]. Consider a domain comprising straight dislocations. Thus, each dislocation can be characterized by a constant sense $\boldsymbol{\xi}$ and its Burgers vector $\boldsymbol{b}$. We can then divide all dislocations into sets $\mathcal{S}$ according to these properties . Each set thus has a corresponding density ρ. The Kröner–Nye tensor is then defined via

$$\boldsymbol{\alpha} = \sum_{\mathcal{S}} \rho_{\mathcal{S}} \boldsymbol{b}_{\mathcal{S}} \otimes \hat{\boldsymbol{\xi}}_{\mathcal{S}}. \tag{2.26}$$

One property of the Kröner–Nye tensor is that it yields the effective deformation character of an ensemble of dislocations. Assume two sets of dislocation, both with the same density and Burgers vector but opposite senses. The resulting Kröner–Nye tensor is zero. Similarly, two sets having the same density and sense but opposite Burgers vectors yield zero. This is why the Kröner–Nye tensor can be seen as a density-like measure for geometrically necessary dislocations (GNDs).

2.2.5.3. Higher-dimensional dislocation density

In contrast to aforementioned dislocation density measures, Hochrainer, Zaiser, and Gumbsch [21] propose a dislocation density measure that not only depends on the position, but the dislocation line orientation within the slip system as well. This means that parts of dislocations that are close to each other in three-dimensional space are not necessarily close in the orientation-dimension. When separate densities for different slip systems are defined, it allows taking the local character of the dislocation into account, which – as mentioned in section 2.2.1 – depends on the orientation of the tangent vector $\hat{\boldsymbol{\xi}}$ with respect to the Burgers vector $\boldsymbol{b}$. This is important for modeling materials that exhibit different behavior of the dislocations depending on their local character, e.g., the velocity they move at for a given Peach–Köhler force or whether cross-slip is possible.

2.2.5.4. Dislocation density alignment tensors

The higher-dimensional dislocation density can be expanded into an infinite hierarchy of dislocation density alignment tensors [22]

$$\boldsymbol{\rho}^{(n)}(\boldsymbol{x}) = \int_0^{\pi} \int_0^{2\pi} \rho(\boldsymbol{x}, \theta, \varphi)\, \hat{\boldsymbol{\xi}}(\theta, \varphi)^{\otimes n} \,\mathrm{d}\varphi\, \mathrm{d}\theta, \tag{2.27}$$

where $\bullet^{\otimes n}$ denotes the tensor power

$$\hat{\boldsymbol{\xi}}^{\otimes n} = \underbrace{\hat{\boldsymbol{\xi}} \otimes \cdots \otimes \hat{\boldsymbol{\xi}}}_{n \text{ times}}, \tag{2.28}$$

and θ and φ are the polar angle and the azimuthal angle of the dislocation orientation space, respectively. We also refer to the dislocation density alignment tensor fields as CDD fields. As is, the character of the dislocation is not taken into account in its entirety, as the Burgers vector $\boldsymbol{b}$ is missing. This can be remedied by computing the higher-dimensional dislocation density and subsequently the dislocation density alignment tensors separately for each slip system. As each slip system is also characterized by a normal vector $\boldsymbol{n}$, the polar angle θ may be dropped if a two-dimensional coordinate system on the slip system is chosen. Furthermore, if the two-dimensional coordinate system is chosen such that the Burgers vector $\boldsymbol{b}$ is parallel to the first axis, components of the dislocation density alignment tensor can be interpreted as screw and edge components of the dislocation. The tangent vector can then be written as

$$\hat{\boldsymbol{\xi}}(\varphi) = \begin{pmatrix} \cos\varphi \\ \sin\varphi \end{pmatrix} = \begin{pmatrix} \xi_{\text{screw}} \\ \xi_{\text{edge}} \end{pmatrix}. \tag{2.29}$$

In the following, the interpretation of low-order dislocation density alignment tensors is given. We provide a figure for each order that shows the respective fields for three different dislocation microstructures. In all cases is the Burgers vector of all dislocations parallel to the x-axis. The first dislocation microstructure is a single dislocation. We use this to show the information extracted by each dislocation density alignment tensor along a single curve. Next, we provide a microstructure comprising randomly placed dislocation loops with random sense. This emulates a microstructure of statistically stored dislocations (SSDs). Finally, we consider a so-called edge dislocation wall. This particular dislocation configuration is part of persistent slip bands, a dislocation microstructure observed in fatigue experiments [37]. The “walls” comprise straight edge dislocations. Overall, the edge dislocations in the walls are balanced and could be considered as SSDs. Few dislocations bow out of this wall and most density is concentrated in the walls.

The 0th-order dislocation density alignment tensor

$$\rho^{(0)} = \int_0^{\pi}\int_0^{2\pi} \rho(\boldsymbol{x}, \theta, \varphi)\,\mathrm{d}\varphi\,\mathrm{d}\theta \tag{2.30}$$

is the total dislocation density ρ^{t}, which we will henceforth denote using the symbol $\rho^{(0)}$. Another field used within the CDD framework is the dislocation curvature density $q^{(0)}$, which enables us to take the average curvature $\langle k\rangle$ into account. The latter may be computed via $\langle k\rangle = q^{(0)}/\rho^{(0)}$. Both 0th-order CDD fields are shown in Figure 2.5. Along a single dislocation, $\rho^{(0)}$ is constant. As

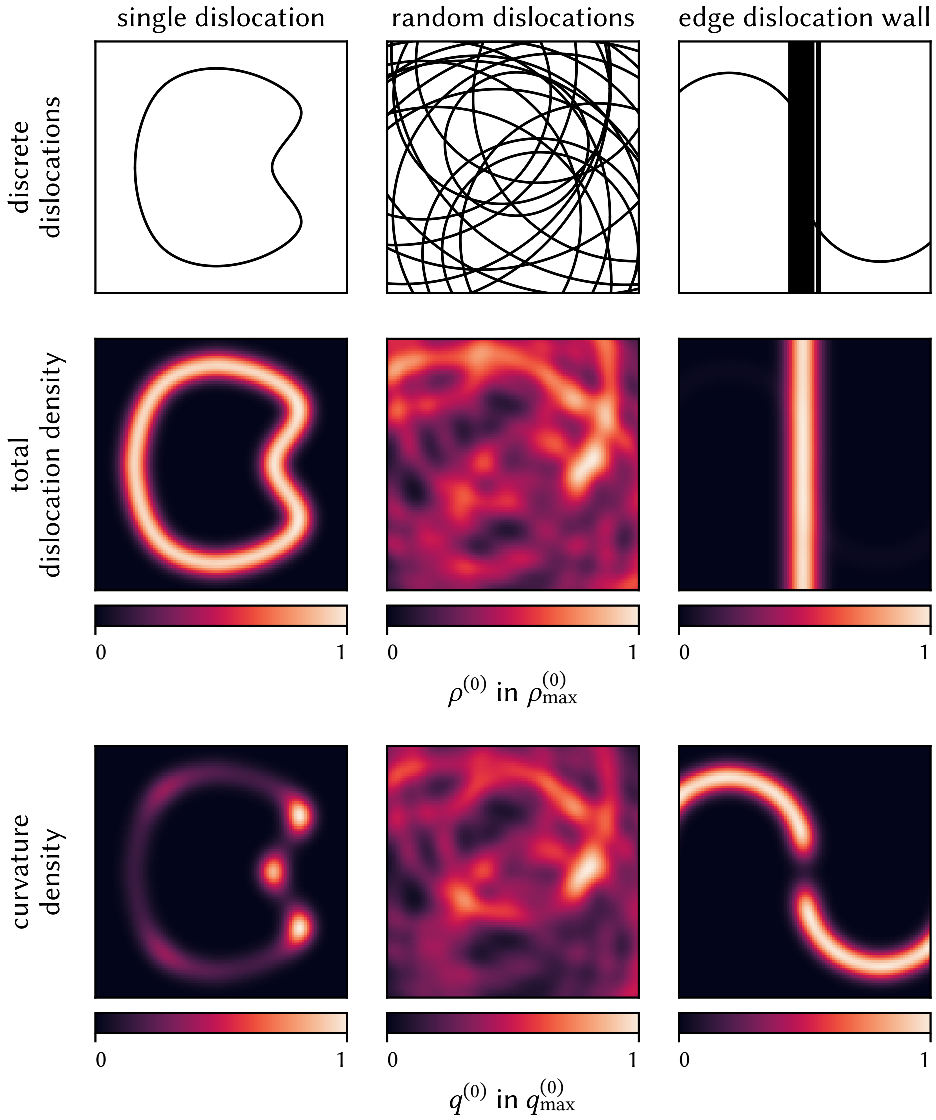


Figure 2.5.: 0^{th}-order CDD fields for different dislocation microstructures.

several dislocations overlap, it reveals the positions where most dislocations are within a limited distance to each other. For the edge dislocation wall, we can see that the density contribution of the bowing out dislocations is negligible compared to the one of the edge dislocations forming the wall. The dislocation curvature density $q^{(0)}$ similarly reveals the positions where dislocations exhibit a large average curvature. For the single dislocation, these are the dents on the right-hand side. For the edge dislocation wall, it emphasizes the bowing out dislocations that are hardly perceptible with $\rho^{(0)}$. The reason for this is that the straight edge dislocations have zero curvature and therefore do not contribute to $\rho^{(0)}$ at all.

While these interpretations are independent of the coordinate system, the following interpretations of the dislocation density alignment tensors are only valid within a slip system specific coordinate system and separate higher-dimensional dislocation densities for every slip system.

The next order

$$\boldsymbol{\rho}^{(1)} = \int_0^{\pi} \int_0^{2\pi} \rho(\boldsymbol{x}, \theta, \varphi)\, \hat{\boldsymbol{\xi}}(\theta, \varphi)\, \mathrm{d}\varphi\, \mathrm{d}\theta \tag{2.31}$$

then represents the excess dislocation density, sometimes also referred to as density of GNDs. They are shown in Figure 2.6. As the Burgers vector is parallel to the $\boldsymbol{x}$, parts of the dislocation line that are parallel it contribute to the screw type GND density, whereas parts perpendicular to it contribute to the edge type GND density. Note that the sign of the density depends on the orientation of the sense with respect to the coordinate axes. For the single dislocation, every part of the dislocation counts as geometrically necessary. This is resembled in range of values $\boldsymbol{\rho}^{(1)}$ takes on, i.e., $-0.95\,\rho^{(0)}_{\mathrm{max}}$ to $0.95\,\rho^{(0)}_{\mathrm{max}}$. On the contrary, only about 50 % of the total dislocation density of the random dislocation arrangement count as GND density. The edge dislocation wall configuration shows the bowing out dislocation parts clearly, as the bulk of dislocations inside the wall is of edge type. These edge dislocations are mostly SSD, resulting in only about $-0.15\,\rho^{(0)}_{\mathrm{max}}$ to $0.15\,\rho^{(0)}_{\mathrm{max}}$ being edge GND density. In this configuration, the shortcoming of only using $\rho^{(0)}$ and $\boldsymbol{\rho}^{(1)}$ becomes clear. We know from the discrete structure that the wall mostly comprises pure edge type dislocations. If only aforementioned fields are considered, though, we know via $\rho^{(0)}$ that we have quite a lot of dislocation density in the wall, but via $\boldsymbol{\rho}^{(1)}$ we know that only little of it comes from GNDs with edge character. The only thing the fields tell us about this remaining density is that it comes from SSDs, but nothing regarding the character of dislocations there. This information requires us to additionally include higher-order fields.

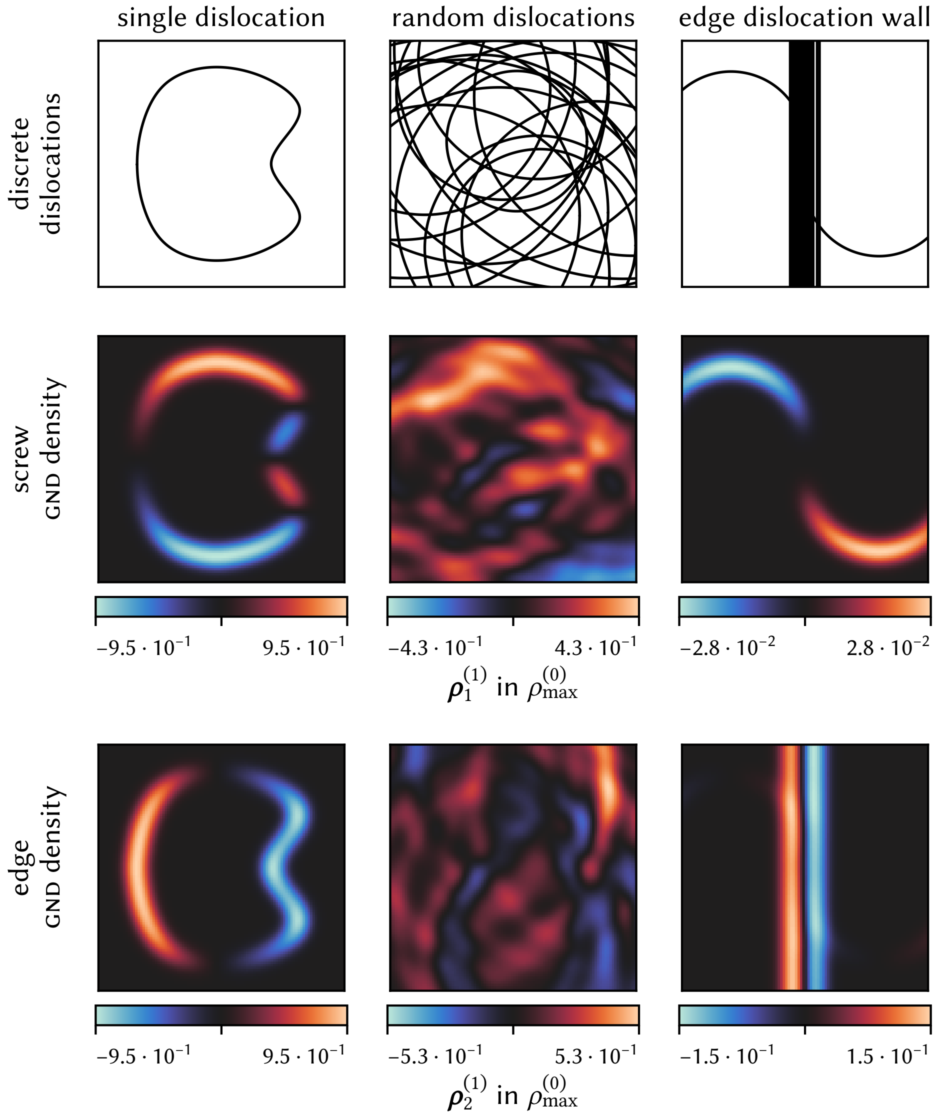


Figure 2.6.: 1st-order CDD fields for different dislocation microstructures.

The next higher-order dislocation density alignment tensor

$$\boldsymbol{\rho}^{(2)} = \int_0^{\pi} \int_0^{2\pi} \rho(\boldsymbol{x}, \theta, \varphi)\, \hat{\boldsymbol{\xi}}(\theta, \varphi) \otimes \hat{\boldsymbol{\xi}}(\theta, \varphi)\, \mathrm{d}\varphi\, \mathrm{d}\theta \tag{2.32}$$

represents the orientation dislocation density, whose main diagonal elements are the orientation specific contributions to the total dislocation density. If a coordinate system is chosen according to (2.29), the main diagonal components correspond to the total screw type dislocation density and total edge type dislocation density. This is shown in Figure 2.7. Like for $\boldsymbol{\rho}^{(1)}$, we can associate components with dislocation parts of screw and edge character, in this case $\boldsymbol{\rho}^{(1)}_{11}$ and $\boldsymbol{\rho}^{(1)}_{22}$, respectively. But unlike $\boldsymbol{\rho}^{(1)}$, they are exclusively positive and direction-specific. The single dislocations shows that parts of the dislocation that are parallel to the x-axis contribute to the screw component and parts that are perpendicular contributing to the edge component. Off-diagonal components of $\boldsymbol{\rho}^{(2)}$ contain information about which orientation mixed type dislocation have. Negative values denote that the dislocation part is oriented from upper left to lower right, or vice-versa. Values are positive when it is oriented from upper right to lower left, or vice-versa.

Once we consider $\boldsymbol{\rho}^{(2)}$ for the dislocation wall structure in addition to the information contained in $\rho^{(0)}$ and $\boldsymbol{\rho}^{(1)}$, we can infer the character of dislocation within the walls. The component of $\boldsymbol{\rho}^{(2)}$ that contributes the most to the dislocation wall is the one that represents edge dislocations. We can therefore conclude that the SSDs inside the wall are of edge character. Thus, adding $\boldsymbol{\rho}^{(2)}$ enables us to characterize this structure accurately from continuum fields alone.

An important observation is that the main diagonal components of $\boldsymbol{\rho}^{(2)}$ add up to the total dislocation density $\rho^{(0)}$. This is a part of a more general relationship between different even-order and between different odd-order dislocation density alignment tensors [22]

$$\operatorname{tr}\big(\boldsymbol{\rho}^{(n)}\big) = \boldsymbol{\rho}^{(n-2)}, \tag{2.33}$$

where $\operatorname{tr} \bullet$ denotes the trace. This relationship implies that all information represented by the dislocation density alignment tensors of up to the order m are contained in the tensors $\boldsymbol{\rho}^{(m)}$ and $\boldsymbol{\rho}^{(m-1)}$.

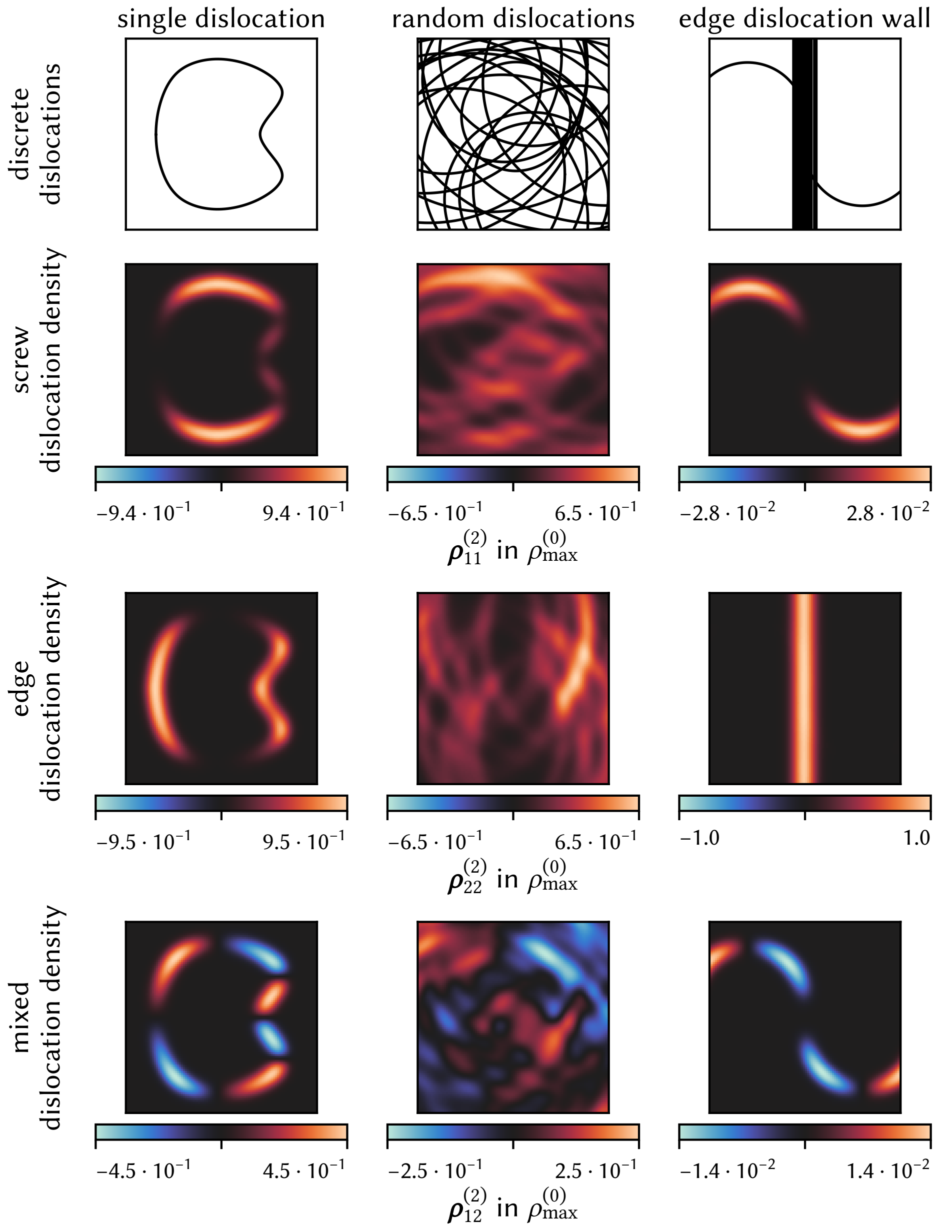


Figure 2.7.: 2nd-order CDD fields for different dislocation microstructures.

3. Methods

This part introduces all methodologies and frameworks that are the basis for the investigations and analyses of the subsequent parts. We first introduce the finite element method, which we apply in the context of linear elasticity to compute internal stresses of a body under external load and with dislocations. Subsequently, we summarize the simulation method of discrete dislocation dynamics. We then introduce the discrete-to-continuous (D2C) method as a means of converting discrete dislocation data to continuous fields. Lastly, we summarize the basics of machine learning, introduce the machine learning algorithms used in this work, and outline how we use aforementioned continuous fields to apply machine learning on dislocation microstructures.

3.1. Finite element method

The finite element method (FEM) is a popular numerical method for solving partial differential equations (PDEs). This is achieved by subdividing an arbitrarily shaped domain into so-called elements that form the finite element mesh. Based on the weak form of the PDEs, a global system of linear equations is assembled using the equations of each element. Subsequently, this global system of linear equations is solved and quantities of interest are computed in a postprocessing step.

In the following, we first outline the derivation of the weak form utilized in this work. We then summarize the steps taken to discretize the problem domain in space and assemble the system of linear equations. These two sections are based mostly on Fish and Belytschko [38]. We then provide more details for the right-hand side contribution of the dislocation-based eigenstrain. Finally, we briefly summarize the method used in this work to adaptively refine the finite element mesh to increase the accuracy of the numerical solution while keeping the computational cost relatively low. The actual implementation of the FEM was done in deal.II version 9.0 [39, 40].

3.1.1. Weak form

We neglect body forces, i.e., $\boldsymbol{B} = \boldsymbol{0}$. The weak form for linear elastic problems is obtained by first multiplying the balance of linear momentum equation (2.5) with an admissible, smooth weight function $\boldsymbol{w}$ and integrating over the domain Ω, resulting in

$$\int_{\Omega} \boldsymbol{w} \cdot \operatorname{div} \boldsymbol{\sigma} \, \mathrm{d}V = 0. \tag{3.1}$$

Note that by “admissible” we mean that only weight functions that vanish on the boundaries where displacements are prescribed permitted. Subsequently, Green’s theorem

$$\int_{\Omega} \boldsymbol{w} \cdot \operatorname{div} \boldsymbol{\sigma} \, \mathrm{d}V = \int_{\Gamma} \boldsymbol{w} \cdot \boldsymbol{\sigma} \cdot \boldsymbol{n} \, \mathrm{d}A - \int_{\Omega} \nabla_{\mathrm{S}} \boldsymbol{w} : \boldsymbol{\sigma} \, \mathrm{d}V \tag{3.2}$$

is applied to yield

$$\int_{\Omega} \nabla_{\mathrm{S}} \boldsymbol{w} : \boldsymbol{\sigma} \, \mathrm{d}V = \int_{\Gamma} \boldsymbol{w} \cdot \boldsymbol{\sigma} \cdot \boldsymbol{n} \, \mathrm{d}A. \tag{3.3}$$

Within the FEM, two boundary conditions are distinguished that result in dividing the entirety of the boundary Γ into two disjoint parts; essential boundary conditions and natural boundary conditions. For linear elasticity, they typically refer to the prescription of displacements

$$\boldsymbol{u} = \bar{\boldsymbol{u}} \quad \text{on} \quad \Gamma_{\boldsymbol{u}}, \tag{3.4}$$

and the prescription of tractions

$$\boldsymbol{\sigma} \cdot \boldsymbol{n} = \bar{\boldsymbol{t}} \quad \text{on} \quad \Gamma_{\boldsymbol{t}}, \tag{3.5}$$

respectively. Due to our choice of weight functions, the displacement boundary conditions are always fulfilled exactly and therefore do not have to be considered in the following.

Substituting the traction boundary condition into equation (3.3) yields

$$\int_{\Omega} \nabla_{\mathrm{S}} \boldsymbol{w} : \boldsymbol{\sigma} \, \mathrm{d}V = \int_{\Gamma_{\boldsymbol{t}}} \boldsymbol{w} \cdot \bar{\boldsymbol{t}} \, \mathrm{d}A. \tag{3.6}$$

Applying the constitutive law equation (2.7) yields

$$\int_{\Omega} \nabla_{\mathrm{S}} \boldsymbol{w} : \boldsymbol{C} : \boldsymbol{\varepsilon} \, \mathrm{d}V = \int_{\Gamma_{\boldsymbol{t}}} \boldsymbol{w} \cdot \bar{\boldsymbol{t}} \, \mathrm{d}A. \tag{3.7}$$

We then take the eigenstrain contributions of dislocations into account by applying equation (2.22) and bringing the respective term to the right-hand side

$$\int_{\Omega} \nabla_{\mathrm{S}} \boldsymbol{w} : \boldsymbol{C} : \boldsymbol{\varepsilon}^{\mathrm{tot}} \, \mathrm{d}V = \int_{\Gamma_t} \boldsymbol{w} \cdot \bar{\boldsymbol{t}} \, \mathrm{d}A + \int_{\Omega} \nabla_{\mathrm{S}} \boldsymbol{w} : \boldsymbol{C} : \boldsymbol{\varepsilon}^{\mathrm{dis}} \, \mathrm{d}V. \tag{3.8}$$

Finally, we use the relationship of displacements to strains given in equation (2.4) to arrive at the weak form

$$\int_{\Omega} \nabla_{\mathrm{S}} \boldsymbol{w} : \boldsymbol{C} : \nabla_{\mathrm{S}} \boldsymbol{u} \, \mathrm{d}V = \int_{\Gamma_t} \boldsymbol{w} \cdot \bar{\boldsymbol{t}} \, \mathrm{d}A + \int_{\Omega} \nabla_{\mathrm{S}} \boldsymbol{w} : \boldsymbol{C} : \boldsymbol{\varepsilon}^{\mathrm{dis}} \, \mathrm{d}V. \tag{3.9}$$

3.1.2. Spatial discretization

The problem domain is discretized into n_{el} elements. We may then rewrite the integral over the whole domain and boundary given in equation (3.9) as the sum of the integrals over each element domain Ω^e and boundary Γ^e:

$$\begin{aligned} &\sum_{e=1}^{n_{\mathrm{el}}} \left\{ \int_{\Omega^e} \nabla_{\mathrm{S}} \boldsymbol{w} : \boldsymbol{C} : \nabla_{\mathrm{S}} \boldsymbol{u} \, \mathrm{d}V \right\} \\ &\qquad = \sum_{e=1}^{n_{\mathrm{el}}} \left\{ \int_{\Gamma_t^e} \boldsymbol{w} \cdot \bar{\boldsymbol{t}} \, \mathrm{d}A + \int_{\Omega^e} \nabla_{\mathrm{S}} \boldsymbol{w} : \boldsymbol{C} : \boldsymbol{\varepsilon}^{\mathrm{dis}} \, \mathrm{d}V \right\}. \end{aligned} \tag{3.10}$$

Each element comprises n_{en} nodes. We may then compute the displacements within an element by interpolating an approximation of the displacements defined at each node

$$\boldsymbol{u}(\boldsymbol{x}) \approx \boldsymbol{N}^e(\boldsymbol{x}) \cdot \tilde{\boldsymbol{u}}^e, \quad \boldsymbol{x} \in \Omega^e, \tag{3.11}$$

where $\boldsymbol{N}^e$ denotes the shape functions. Within this work, we use linear Lagrange polynomials for the interpolation of nodal values within an element. In the same way, the weight functions are interpolated as well.

The mapping of the element nodal displacement approximation $\tilde{\boldsymbol{u}}^e$ to the global displacement approximation $\tilde{\boldsymbol{u}}$ is done via a so-called gather operator $\boldsymbol{L}^e$

$$\tilde{\boldsymbol{u}}^e = \boldsymbol{L}^e \cdot \tilde{\boldsymbol{u}}, \tag{3.12}$$

which we also use for the weight function. Upon inserting this in equation (3.10), we get

$$\begin{aligned}\tilde{\boldsymbol{w}} \cdot \sum_{e=1}^{n_{\text{el}}} \boldsymbol{L}^{eT} \cdot \int\limits_{\Omega^e} (\nabla_{\text{S}} \boldsymbol{N}^e)^T : \boldsymbol{C} : \nabla_{\text{S}} \boldsymbol{N}^e \, \mathrm{d}V \cdot \boldsymbol{L}^e & \\ = \tilde{\boldsymbol{w}} \cdot \sum_{e=1}^{n_{\text{el}}} \boldsymbol{L}^{eT} \cdot \left(\int\limits_{\Gamma_t^e} \boldsymbol{N}^{eT} \cdot \bar{\boldsymbol{t}} \, \mathrm{d}A + \int\limits_{\Omega^e} (\nabla_{\text{S}} \boldsymbol{N}^e)^T : \boldsymbol{C} : \boldsymbol{\varepsilon}^{\text{dis}} \, \mathrm{d}V \right). & \quad (3.13)\end{aligned}$$

This relationship should be true for all $\tilde{\boldsymbol{w}}$. Thus, the two sums have to be equal.

We can simplify this equation by introducing the contribution of an element to the matrix of the system of linear equations

$$\boldsymbol{K}^e = \int\limits_{\Omega^e} (\nabla_{\text{S}} \boldsymbol{N}^e)^T : \boldsymbol{C} : \nabla_{\text{S}} \boldsymbol{N}^e \, \mathrm{d}V \tag{3.14}$$

and the contribution of an element to the right-hand side of the system of linear equations

$$\boldsymbol{f}^e = \int\limits_{\Gamma_t^e} \boldsymbol{N}^{eT} \cdot \bar{\boldsymbol{t}} \, \mathrm{d}A + \int\limits_{\Omega^e} (\nabla_{\text{S}} \boldsymbol{N}^e)^T : \boldsymbol{C} : \boldsymbol{\varepsilon}^{\text{dis}} \, \mathrm{d}V \tag{3.15}$$

to yield

$$\left(\sum_{e=1}^{n_{\text{el}}} \boldsymbol{L}^{eT} \cdot \boldsymbol{K}^e \cdot \boldsymbol{L}^e \right) \cdot \tilde{\boldsymbol{u}} = \sum_{e=1}^{n_{\text{el}}} \boldsymbol{L}^{eT} \cdot \boldsymbol{f}^e. \tag{3.16}$$

Written in terms of the global system, we can see the final matrix form of the system of linear equations

$$\boldsymbol{K} \cdot \tilde{\boldsymbol{u}} = \boldsymbol{f}. \tag{3.17}$$

However, what remains are the integrations in equations (3.14) and (3.15). A popular method of determining them is via Gaussian quadrature. Instead of analytically computing the integral, the value of the integrand is computed at n_{gp} specific points $\boldsymbol{x}_i$. These values are multiplied with point-specific weights w_i and summed up. Thus, equations (3.14) and (3.15) may be written as

$$\boldsymbol{K}^e \approx \sum_{i=1}^{n_{\text{gp}}^{\Omega^e}} \left(\nabla_{\text{S}} \boldsymbol{N}^e(\boldsymbol{x}_i) \right)^T : \boldsymbol{C} : \nabla_{\text{S}} \boldsymbol{N}^e(\boldsymbol{x}_i) w_i, \quad \text{and} \tag{3.18}$$

$$\boldsymbol{f}^e \approx \sum_{i=1}^{n_{\text{gp}}^{\Gamma^e}} \boldsymbol{N}^e(\boldsymbol{x}_i)^T \cdot \bar{\boldsymbol{t}}(\boldsymbol{x}_i) w_i + \sum_{i=1}^{n_{\text{gp}}^{\Omega^e}} \left(\nabla_{\text{S}} \boldsymbol{N}^e(\boldsymbol{x}_i) \right)^T : \boldsymbol{C} : \boldsymbol{\varepsilon}^{\text{dis}}(\boldsymbol{x}_i) w_i. \tag{3.19}$$

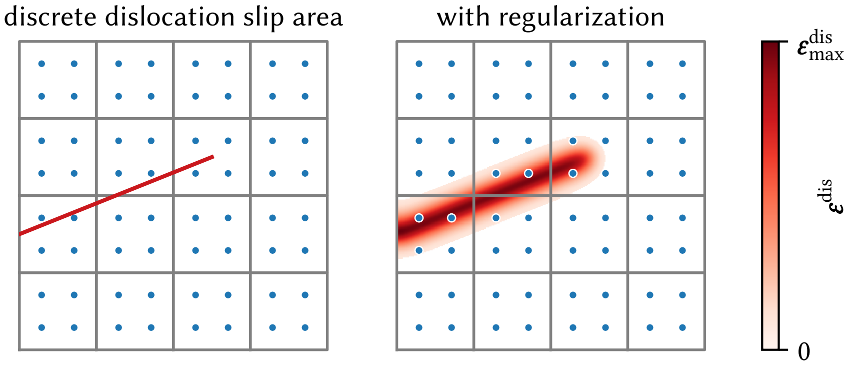


Figure 3.1.: A FEM mesh with a dislocation slip area (red) without (left) and with (right) regularization. Quadrature points are shown in blue.

The right-hand side term containing the contribution of the tractions is straightforward to compute, as the points used for the Gaussian quadrature lie on the boundary that the prescribed traction $\bar{\boldsymbol{t}}$ is defined on. However, the term containing the contribution of the dislocations requires special care.

3.1.3. Implementation of the dislocation eigenstrain

The issue with computing the eigenstrain contribution of dislocations $\boldsymbol{\varepsilon}^{\mathrm{dis}}$ within the Gaussian quadrature is the following: As we evaluate the integrand at points $\boldsymbol{x}_i$, we also compute $\boldsymbol{\varepsilon}^{\mathrm{dis}}$ at these points. However, recall from equation (2.22) that the dislocation eigenstrain is only defined at points that are part of the dislocation slip area. If we do not keep this in mind the chances of actually taking dislocations into account are low, as no point used for the Gaussian quadrature might be part of a dislocation slip area. This can be seen on the left-hand side of Figure 3.1. There are two potential remedies, which we outline in the following.

First, we could generate the finite element mesh in such a way that the integration points of relevant elements align perfectly with the dislocation slip areas. This comes with some limitations. On the one hand, it requires us to use a tailor-made mesh for every computation that we want to conduct. For few dislocations, this might be feasible. But as the number of dislocations grows, both the human effort required and the capabilities of finite element mesh preprocessors make this approach impractical. On the other hand, it means we can not apply adaptive mesh refinement. We therefore have to make a best guess about where we needed a finer mesh resolution to improve the accuracy of our analysis.

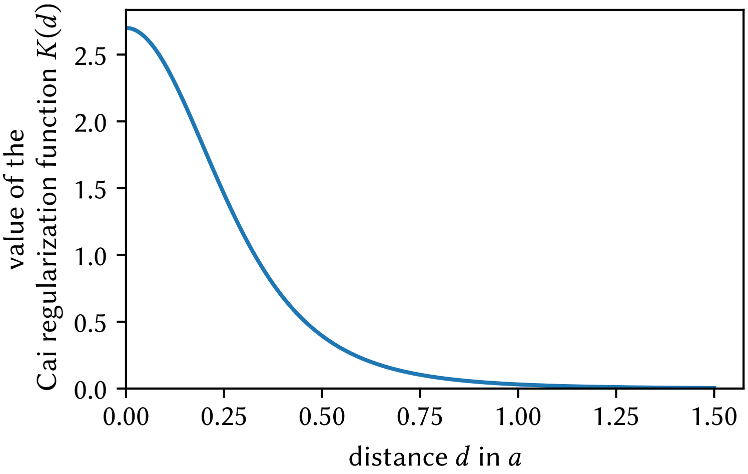


Figure 3.2.: Value of the Cai regularization function K over distance d.

Another approach was suggested by Jamond et al. [41]. Instead of treating the dislocation slip area as a two-dimensional surface, we regularize and represented it by a three-dimensional “area density”. To this end, we use the previously mentioned Burgers vector distribution function proposed by Cai et al. [1]

$$K(d) = \frac{15}{8\pi}\left(\frac{0.3425}{a_1^3\left(d^2/a_1^2+1\right)^{7/2}}+\frac{0.6575}{a_2^3\left(d^2/a_2^2+1\right)^{7/2}}\right), \tag{3.20}$$

with $a_1 = 0.9038a$ and $a_2 = 0.5451a$. A plot of this radially symmetric function is shown in Figure 3.2. On the right-hand side of Figure 3.1, the configuration is shown after applying this regularization method. Due to the dislocation slip area now being spread along all dimensions, several quadrature points in the vicinity of the dislocation slip area exhibit non-zero dislocation eigenstrain contributions. However, special care has to be taken close to boundaries. By spreading the dislocation slip areas we put parts of it outside of the domain. This inconsistency can lead to artificial stresses at the surface close to the dislocation slip area. We follow the approach suggested by Jamond et al. [41], and elongate the dislocation slip areas outside of the domain before applying the regularization. This way, these artificial stresses are not observed. When using this implementation, the same mesh may be used for several different dislocation configurations within the same geometry. Furthermore, it allows the application of adaptive mesh refinement as a means of introducing higher finite element mesh resolution where it improves the accuracy of the solution the most.

3.1.4. Adaptive mesh refinement

The accuracy of solutions obtained via the FEM is sensitive to the spatial discretization. In general, a balance is struck between the increase in both accuracy and computational cost when conducting FEM analyses. A typical technique is to use gradual meshes that are finer at regions of interest, e.g., in the vicinity of a notch or crack where large stress gradients are expected. An alternative approach is to compute the solution for a rather coarse mesh, and in a postprocessing step determine an approximation of the error within each element. Subsequently, the elements containing the largest errors are refined, and the computation is run again. These steps may be repeated for a fixed amount of times or until a maximum number of elements is reached. This process is called adaptive mesh refinement and is readily implemented in deal.II.

The common parameters include the error estimator which is used to decide on elements to refine, the amount of elements to refine per step, and potentially also which previously refined elements to coarsen again. In this work, we use a modified version of the error estimator proposed by Kelly et al. [42] to decide which elements to refine. This error estimator is already implemented in the deal.II framework.

It works by computing the differences between solution value derivatives obtained at shared faces between elements. First, all elements sharing faces with an element are determined. Then, the difference of the solution derivatives across each face is integrated and summed up for each element. Finally, elements that exhibit large mismatches between its solution derivatives and the solution derivatives of its adjacent elements get refined.

3.2. Discrete dislocation dynamics

Discrete dislocation dynamics (DDD) are a numerical method for evolving dislocation microstructures in time [43, 44, 45, 46]. Within DDD, dislocations are treated as discretized curves. Different ways or representing these curves exist, and in the following we introduce the ones found in the DDD framework used throughout this work.

The first way is by using line segments of fixed orientation and length, which form a discrete lattice along which dislocations move. This representation allows efficient computation of dislocation-dislocation interactions as stresses exerted by the limited set of segment types can be precomputed. However, it is the crudest approach of representing dislocations and the possible segment orientations have to be selected carefully as not to neglect important dislocation reactions. In this work, we make use of the microMegas code [47], which is based this kind of representation.

A more flexible approach is representing dislocation curves by a sequence of nodes, i.e., as polygonal chains. The line segments connecting these nodes can be of arbitrary orientation, therefore offering more flexibility than the previously mentioned approach. As no precomputation of dislocation stress fields is possible, it is also computationally more expensive. The DDD code of Daniel Weygand [45, 48] – which is also used in this work – makes use of this representation.

Finally, the most faithful approximation being used by current DDD codes is the discretization of dislocation lines via splines. In addition to arbitrary orientations, this representation also allows to accurately represent the curvature along the dislocation. Out of the methods of representing dislocations, this is the numerically most expensive one. A DDD code used in the present work which uses this representation is the MODEL code [49, 46].

The driving force of dislocation motion in DDD is the Peach–Köhler force, see equation (2.13). It is integrated along the discretization of the dislocation line and used to determine the velocity of the dislocations via a mobility law. A commonly used mobility law for FCC materials assumes that the velocity of the dislocations is proportional to the Peach–Köhler force acting upon them [50, 47]. Sources of stresses within DDD are, e.g., external loading, dislocation-surface interactions and dislocation-dislocation interactions.

Dislocation reactions like, e.g., dislocation annihilation, cross-slip, and the formation and unzipping of junctions are implemented using so-called "local rules". These are phenomenological rules informed by experiments or simulations on lower length-scales, e.g., atomistics or ab initio calculations. By turning local rules on and off, the impact of a dislocation reaction on the dislocation microstructure and overall deformation behavior of a material can be studied.

3.3. Extraction of dislocation microstructure features

3.3.1. The discrete-to-continuous method

Sandfeld and Po [51] introduced the D2C method to validate the kinematics of Hochrainer's CDD theory, i.e., whether the evolution equations are actually able to accurately represent the motion of line densities. By converting results of DDD simulations into the dislocation density fields used by the CDD theory, the D2C method enabled the quantitative comparison between those results.

Within the D2C method, we treat dislocations as parametrized curves $\mathscr{C}(t)$, where $t \in [a, b]$ denotes the curve parameter. This representation enables the convenient computation of measures required for the determination of the continuum fields. In this work, we make use of the tangent vector

$$\hat{\xi}(t) = \frac{\mathrm{D}_t\,\mathscr{C}(t)}{\|\mathrm{D}_t\,\mathscr{C}(t)\|}, \tag{3.21}$$

and the unsigned curvature

$$k(t) = \frac{\sqrt{\|\mathrm{D}_t\,\mathscr{C}(t)\|^2 \|\mathrm{D}_t^2\,\mathscr{C}(t)\|^2 - \left(\mathrm{D}_t\,\mathscr{C}(t) \cdot \mathrm{D}_t^2\,\mathscr{C}(t)\right)^2}}{\|\mathrm{D}_t\,\mathscr{C}(t)\|^3}, \tag{3.22}$$

where $\mathrm{D}_t^n\,\mathscr{C}(t)$ denotes the n^{th} derivative of the curve $\mathscr{C}(t)$ with respect to its curve parameter t. In addition to its spatial configuration, we also associate each curve with the slip plane $\boldsymbol{n}$ and Burgers vector $\boldsymbol{b}$ of the dislocation.

A domain Ω is discretized into subdomains Ω_i. Within each subdomain, we can compute averaged measures of the dislocations intersecting the subdomain via

$$\frac{1}{V_{\Omega_i}} \sum_{\mathscr{C}} \int\limits_{\mathscr{C} \cap \Omega_i} f_{\mathscr{C}}(t) \|\mathrm{D}_t\,\mathscr{C}(t)\|\,\mathrm{d}t, \tag{3.23}$$

where $f_{\mathscr{C}}(t)$ is a function whose expression depends on the measure we intend to compute. In each subdomain, only the parts of the dislocation intersecting the subdomain $\mathscr{C} \cap \Omega_i$ are used for the computation of the measure within the subdomain. An overview of measures used within this work, as well as their expression for $f_{\mathscr{C}}(t)$ are found in Table 3.1.

In section 2.2.5.4, we outlined that the dislocation density alignment tensors do not take the Burgers vector into account, and therefore do not capture all physical properties of dislocations by themselves. The Kröner–Nye tensor does not suffer from this. By including the Burgers vector in its definition, it is able to take the deformative character of dislocations into account. But unlike the dislocation density alignment tensors, it can not represent SSDs. The latter achieve this by including a variable tensor power of the tangent vector in their definition. We propose a dislocation feature density tensor

$$\boldsymbol{\rho}^{(m,n)}(\boldsymbol{x}) = \int\limits_0^{\pi} \int\limits_0^{2\pi} \rho(\boldsymbol{x}, \theta, \varphi)\, \boldsymbol{b}^{\otimes m} \otimes \hat{\xi}(\theta, \varphi)^{\otimes n}\, \mathrm{d}\varphi\, \mathrm{d}\theta, \tag{3.24}$$

Table 3.1.: Expressions for $f_{\mathscr{C}}(t)$ used to extract microstructure features via equation (3.23).

Microstructure feature	Symbol	$f_{\mathscr{C}}(t)$
dislocation density alignment tensor	$\boldsymbol{\rho}^{(n)}$	$\hat{\boldsymbol{\xi}}_{\mathscr{C}}(t)^{\otimes n}$
dislocation curvature density	$q^{(0)}$	$k_{\mathscr{C}}(t)$
Kröner–Nye tensor	$\boldsymbol{\alpha}$	$\boldsymbol{b}_{\mathscr{C}} \otimes \hat{\boldsymbol{\xi}}_{\mathscr{C}}(t)$
dislocation feature density tensor	$\boldsymbol{\rho}^{(m,n)}$	$\boldsymbol{b}_{\mathscr{C}}^{\otimes m} \otimes \hat{\boldsymbol{\xi}}_{\mathscr{C}}(t)^{\otimes n}$

that combines “the best of both worlds”, i.e., taking the deformative characteristics and spatial configuration of dislocations into account. It represents a generalization of both, dislocation density alignment tensors $\boldsymbol{\rho}^{(n)} = \boldsymbol{\rho}^{(0,n)}$ and the Kröner–Nye tensor $\boldsymbol{\alpha} = \boldsymbol{\rho}^{(1,1)}$.

In this work, we convert data obtained from experiments and DDD simulations into continuum fields to make their features comparable in a quantitative manner. However, only DDD frameworks that use splines to represent the dislocation lines may be directly used within the D2C framework. In other cases, dislocation lines are typically described by polygonal chains, connected series of line segments that are represented by a sequence of vertices. Thus, an important step is to find parametrized curves that best represent the polygonal chains obtained from experiments and simulations.

3.3.2. From polygonal chains to parametrized curves

The polygonal chains extracted from experiments or used in simulations are approximations of the real dislocation line. We therefore try to find parametrized curves that best represent the original dislocation lines based on an intermediate approximation provided as polygonal chains. *How* these polygonal chains were created is important for making a decision on which algorithm to use for the curve approximation.

On the one hand, let us assume we are given a TEM image of a dislocation. When we extract the polygonal chain, we likely do so by placing vertices *exactly* on the dislocation line. The vertices of the polygonal chain should therefore receive more weight, i.e., importance, during the determination of a parametrized curve approximation. On the other hand, within DDD simulations, polygonal chain approximations of the dislocation lines are moved by integrating their driving forces along the line segments making up the polygonal chain. We may therefore assume, that the polygonal chains obtained from

DDD take a more balanced approach, i.e., the vertices and the points on the line segments should be given equal weight.

Eventually, we determine the parametrized curves in two steps. First, we preprocess the polygonal chains depending on whether their dislocation line approximation put more emphasis on the vertices or on both, the vertices and line segments. This results in a polygonal chains with a finer discretization that is then used to fit B-splines. We use these as the parametrized curves for the D2C method.

Rossignac [52] covers three curve subdivision algorithms that work by introducing additional vertices $\boldsymbol{v}_i^{\text{new}} = 0.5(\boldsymbol{v}_i + \boldsymbol{v}_{i+1})$ at the midpoints of the line segments of a polygon and modifying the positions of the original and/or the new vertices.

For the B-spline subdivision, only the original vertices are modified. The new positions $\boldsymbol{v}'$ of the original vertices are computed via

$$\begin{aligned}
\boldsymbol{v}_i' &= \boldsymbol{v}_i + 0.5\big(0.5(\boldsymbol{v}_{i-1}^{\text{new}} + \boldsymbol{v}_i^{\text{new}}) - \boldsymbol{v}_i\big) && (3.25\text{a})\\
&= 0.125(\boldsymbol{v}_{i-1} + 6\boldsymbol{v}_i + \boldsymbol{v}_{i+1}) && (3.25\text{b})\\
\boldsymbol{v}_i^{\text{new}\prime} &= \boldsymbol{v}_i^{\text{new}} && (3.25\text{c})\\
&= 0.5(\boldsymbol{v}_i + \boldsymbol{v}_{i+1}) && (3.25\text{d})
\end{aligned}$$

i.e., they are shifted halfway towards the midpoint of the adjacent new vertices. This algorithm therefore puts all emphasis on the original position of the line segments. Repeated application results in a polygon that converges to a uniform cubic B-spline [52].

For the 4-point subdivision, only the new vertices are shifted from their initial positions. It is therefore the opposite extreme to the B-spline subdivision, and puts all emphasis on the original vertices. The modified positions of the new vertices may be computed via

$$\begin{aligned}
\boldsymbol{v}_i' &= \boldsymbol{v}_i && (3.26\text{a})\\
\boldsymbol{v}_i^{\text{new}\prime} &= \boldsymbol{v}_i^{\text{new}} + 0.25\big(\boldsymbol{v}_i^{\text{new}} - 0.5(\boldsymbol{v}_{i-1}^{\text{new}} + \boldsymbol{v}_{i+1}^{\text{new}})\big) && (3.26\text{b})\\
&= 0.0625(-\boldsymbol{v}_{i-1} + 9\boldsymbol{v}_i + 9\boldsymbol{v}_{i+1} - \boldsymbol{v}_{i+2}) && (3.26\text{c})
\end{aligned}$$

i.e., they are shifted a quarter away from the midpoint of their 2^{nd}-degree neighbors.

The so-called “Jarek’s subdivision” is the middle ground between the B-spline subdivision and the 4-point subdivision. Both, the original and the new vertices are shifted by half the amount the would be shifted in the other two subdivisions. We can therefore compute the new positions via

$$\begin{aligned} \boldsymbol{v}_i' &= \boldsymbol{v}_i + 0.25\big(0.5(\boldsymbol{v}_{i-1}^{\text{new}} + \boldsymbol{v}_i^{\text{new}}) - \boldsymbol{v}_i\big) && \text{(3.27a)} \\ &= 0.0625(\boldsymbol{v}_{i-1} + 14\boldsymbol{v}_i + \boldsymbol{v}_{i+1}) && \text{(3.27b)} \\ \boldsymbol{v}_i^{\text{new}\prime} &= \boldsymbol{v}_i^{\text{new}} + 0.125\big(\boldsymbol{v}_i^{\text{new}} - 0.5(\boldsymbol{v}_{i-1}^{\text{new}} + \boldsymbol{v}_{i+1}^{\text{new}})\big) && \text{(3.27c)} \\ &= 0.03125(-\boldsymbol{v}_{i-1} + 17\boldsymbol{v}_i + 17\boldsymbol{v}_{i+1} - \boldsymbol{v}_{i+2}) && \text{(3.27d)} \end{aligned}$$

The effect of all outlined subdivision algorithms can be seen in Figure 3.3. In the left column, we can see that the B-spline subdivision results in polygonal chains that “undershoot” the original vertices but follows the line segment midpoints rather well. In the right column, we can see how the polygonal chains resulting from the 4-point subdivision exactly intersect the original vertices, but “overshoot” the line segment midpoints. In the center column, we can see polygonal chains after applying Jarek’s subdivision. They slightly “undershoot” the original vertices and slightly “overshoot” the line segment midpoints.

In this work, we make use of Jarek’s subdivision for dislocation data obtained form DDD simulations whose dislocation line representation is based on line segments or vertices. After repeated application of the appropriate subdivision algorithm, we use the cubic B-spline interpolation routine provided by SciPy version 1.4.1 to obtain the final parametrized curve representation used for the computation of the measures via equation (3.23).

3.3.3. Comparing and averaging dislocation microstructures

One benefit of computing the continuum field representation of dislocation microstructures via the D2C method is that it allows us to compare them or to compute their average. However, there are two conditions to be considered. First, the domains Ω must be the same. Second, the spatial discretization by which we define the subdomains Ω_i within Ω must be the same for all dislocation microstructures. Only then it becomes possible to compare or average them.

Assume we wanted to compare two dislocation microstructures with each other. We would then 1. convert the dislocations from their original representation to curves, 2. discretize Ω into $n_{\Omega_{\text{sub}}}$ subdomains Ω_i, and 3. extract continuous fields of our choice for each dislocation microstructure in each subdomain of our spatial discretization via the D2C method equation (3.23). This results in two sets of fields $\bullet$ and $\circ$ for each of the two dislocation microstructures, where $\bullet_i$ and $\circ_i$ refer to the values in the subdomain Ω_i. These

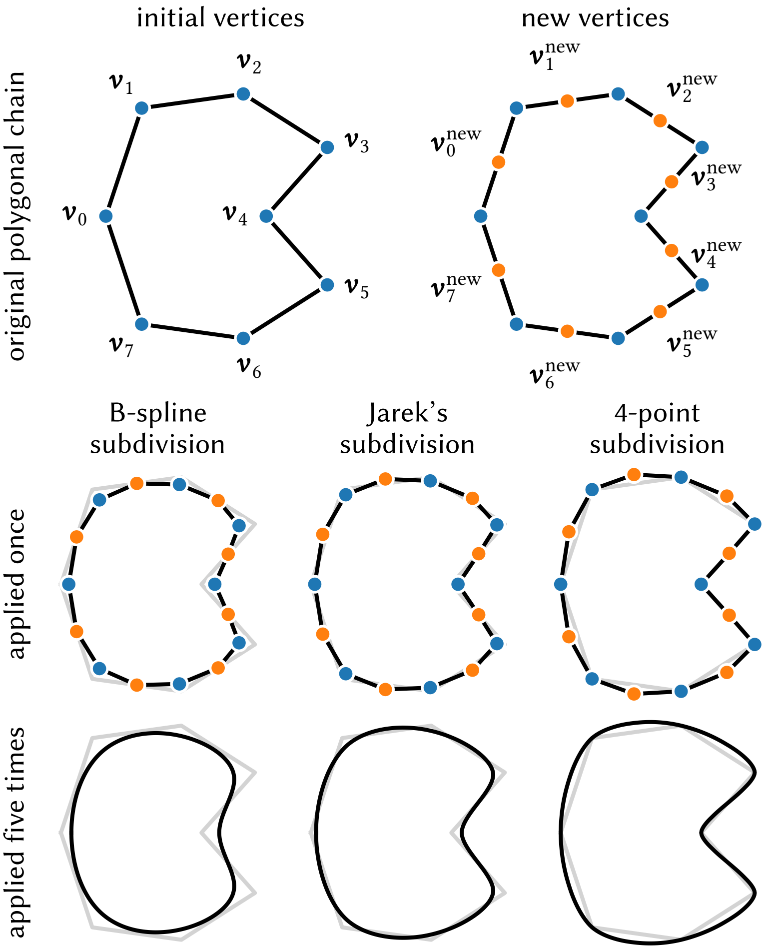


Figure 3.3.: Effect of different curve subdivision algorithms on a polygonal chain. At the top, the original polygonal chain is shown, both with its initial vertices (left) and the new vertices added during the subdivision process (right). The bottom rows show the results after applying different subdivision algorithms once (top) or five times (bottom). To better see the effect of the subdivision algorithms, the initial polygonal chain is shown in gray alongside the result.

values might be scalar quantities like $\rho^{(0)}$ or $q^{(0)}$, tensorial quantities like $\boldsymbol{\rho}^{(1)}$ or $\boldsymbol{\alpha}$, or sets of values like $\{\rho^{(0)}, \boldsymbol{\rho}^{(1)}, \boldsymbol{\rho}^{(2)}, q^{(0)}\}$.

Subsequently, we define a measure of the error $\textsc{e}^{\Omega_\text{sub}}(\bullet_i, \circ_i)$ of the values in a subdomain. In this work, we compute errors only for the scalar 0^th-order dislocation density alignment tensor and use $\textsc{e}^{\Omega_\text{sub}}(\bullet_i, \circ_i) = |\bullet_i - \circ_i|$. However, we are free to define other measures of the error depending on the features of the dislocation microstructures we care most about.

Using this measure of error on the subdomain level, we can then proceed to assign a measure of error $\textsc{e}^{\Omega_\text{sub}}(\bullet, \circ)$ on the domain level. Examples would be the mean absolute error (MAE)

$$\textsc{mae}(\bullet, \circ) = \frac{1}{n_{\Omega_\text{sub}}} \sum_{i=1}^{n_{\Omega_\text{sub}}} \left|\textsc{e}^{\Omega_\text{sub}}(\bullet_i, \circ_i)\right| \tag{3.28}$$

or the root-mean-square error (RMSE)

$$\textsc{rmse}\,(\bullet, \circ) = \sqrt{\frac{1}{n_{\Omega_\text{sub}}} \sum_{i=1}^{n_{\Omega_\text{sub}}} \textsc{e}^{\Omega_\text{sub}}(\bullet_i, \circ_i)^2}. \tag{3.29}$$

If we want to compare the difference of a pair of dislocation microstructures to another pair, these measures are insufficient when the scale of the values between the pairs are different. To facilitate the comparison, we may normalize the domain error by the mean domain averages

$$\langle \bullet \rangle^\Omega = \frac{1}{V_\Omega} \sum_{i=1}^{n_{\Omega_\text{sub}}} \bullet_i V_{\Omega_i} \tag{3.30}$$

and $\langle \circ \rangle^\Omega$, which yields the coefficient of variation (CV) of the MAE

$$\textsc{cv}\big(\textsc{mae}(\bullet, \circ)\big) = \frac{\textsc{mae}(\bullet, \circ)}{0.5\big(\langle \bullet \rangle^\Omega + \langle \circ \rangle^\Omega\big)} \tag{3.31}$$

or the CV of the RMSE

$$\textsc{cv}\big(\textsc{rmse}\,(\bullet, \circ)\big) = \frac{\textsc{rmse}\,(\bullet, \circ)}{0.5\big(\langle \bullet \rangle^\Omega + \langle \circ \rangle^\Omega\big)}. \tag{3.32}$$

These are relative error measures that allow us to better compare the differences between several pairs of dislocation microstructures.

Assume we wanted to determine the average of a set of n_{dm} dislocation microstructure fields $\mathcal{S} = \{\bullet^0, \bullet^1, \dots, \bullet^{n_{\mathrm{dm}}}\}$. We compute the continuum measures we are interested in via the D2C for all dislocation microstructures in every subdomain as we did before. Subsequently, we compute the dislocation microstructure average $\langle \bullet \rangle^{\mathcal{S}}$ by determining the mean of all dislocation microstructures within each subvolume

$$\langle \bullet \rangle_i^{\mathcal{S}} = \frac{1}{n_{\mathrm{dm}}} \sum_{j=1}^{n_{\mathrm{dm}}} \bullet_i^j. \tag{3.33}$$

Besides the average of $\mathcal{S}$, we may also compute the mean absolute deviation (MAD)

$$\textsc{mad}(\mathcal{S}) = \frac{1}{n_{\mathrm{dm}}} \sum_{j=1}^{n_{\mathrm{dm}}} \left| \mathrm{E}^{\Omega}\left(\bullet^j, \langle \bullet \rangle^{\mathcal{S}}\right) \right| \tag{3.34}$$

or the standard deviation of the sample

$$s_{n_{\mathrm{dm}}}^{\mathcal{S}} = \sqrt{\frac{1}{n_{\mathrm{dm}}} \sum_{j=1}^{n_{\mathrm{dm}}} \mathrm{E}^{\Omega}\left(\bullet^j, \langle \bullet \rangle^{\mathcal{S}}\right)^2} \tag{3.35}$$

as measures of the deviation around the mean. As for the measures used for comparison between two dislocation microstructures, comparing the deviations of different sets of dislocation microstructures is only feasible when the scale of their values is similar. By normalizing above measures with the domain average of the dislocation microstructure average we get the CV of the MAD

$$\textsc{cv}\left(\textsc{mad}(\mathcal{S})\right) = \frac{\textsc{mad}(\mathcal{S})}{\left\langle \langle \bullet \rangle^{\mathcal{S}} \right\rangle^{\Omega}} \tag{3.36}$$

and the CV of the standard deviation of the sample

$$\textsc{cv}\left(s_{n_{\mathrm{dm}}}^{\mathcal{S}}\right) = \frac{s_{n_{\mathrm{dm}}}^{\mathcal{S}}}{\left\langle \langle \bullet \rangle^{\mathcal{S}} \right\rangle^{\Omega}}, \tag{3.37}$$

which enable us to compare deviations across different sets of dislocation microstructures.

3.4. Machine learning

One of the main goals of this work is to demonstrate that the various dislocation densities obtained via the D2C method are able to quantify characteristics of dislocation microstructures. We want to do so by using machine learnings to predict the size of a nanoparticle based on the dislocation microstructure that forms during a numerical experiment.

Broadly speaking, machine learning is the practice of using computers to learn from data. Within the data, several observations are described by one or more variables. These variables may be quantitative or qualitative.

If our goal is to learn a function that maps a set of input variables to a set of output variables, it is called supervised learning. Depending on whether the output variable is quantitative or qualitative, we may further divide a supervised learning into regression or classification problems, respectively. If our goal is to learn underlying patterns in the data that are not expressed explicitly in the variables, it is called unsupervised learning. There are other types of machine learning, but most learning problems are either supervised learning or unsupervised learning [53]. In the present work, we deal only with supervised learning problems, for which we outline the fundamental concepts in the following.

In supervised learning, we use a set of n_{obs} observations to find an approximation of a function f. The set of variables of an observation that are used as input of f are denoted by $\boldsymbol{X}$, the (typically scalar) variable that is the output by Y. In general, we can write this relationship as

$$Y = f(\boldsymbol{X}) + \epsilon, \tag{3.38}$$

where ϵ is a random error term whose arithmetic mean is zero [53]. The process of approximating f via fitting a machine learning model is called training, and the set of observations with known input–output pairs we use to train it with is called the training set. Once an approximation $\tilde{f}$ is found, we may use it for prediction and inference.

Prediction means that we use $\tilde{f}$ to compute a predicted output $\check{Y}$ for a given $\boldsymbol{X}$ via

$$\check{Y} = \tilde{f}(\boldsymbol{X}). \tag{3.39}$$

For fixed $\tilde{f}$ and $\boldsymbol{X}$, the expected value of the squared difference between sets of predicted and true output results in two terms [53]

$$\begin{aligned} \mathrm{E}\left[(Y - \check{Y})^2\right] &= \mathrm{E}\left[f(\boldsymbol{X}) + \epsilon - \tilde{f}(\boldsymbol{X})\right] & (3.40) \\ &= \left(f(\boldsymbol{X}) - \tilde{f}(\boldsymbol{X})\right)^2 + \mathrm{Var}\left[\epsilon\right], & (3.41) \end{aligned}$$

where $\mathrm{Var}\left[\epsilon\right]$ denotes the variance of the error term. The first term describes the part of the difference that is reducible with better approximations $\tilde{f}$. The latter term describes an irreducible part of the error done by $\tilde{f}$, meaning that regardless of how well we approximate f, this difference will prevail. One common reason for this error is that the available input variables are not sufficient and could be chosen better. This practice of determining optimal input variables for a given learning problem is called feature engineering.

Inference means that we use $\tilde{f}$ to learn how its output is affected by the input. Assuming that our approximation $\tilde{f}$ of the unknown function f is good, then the former should also have learned the underlying relationships between $\boldsymbol{X}$ and Y. We may then analyze *how* $\tilde{f}$ makes use of the input variables to return an output. Depending on the machine learning algorithm we base our machine learning model on, there might be an analytical expression available that allows interpretation. There are also means of determining how important specific input variable are for $\tilde{f}$ to accurately predict the output, i.e., how to import the feature importance.

The success of both, prediction and inference, rests on how well our machine learning model approximates f. In the following, we first outline how to assess the quality of the approximation of f, i.e., how to determine the accuracy of a machine learning model. Subsequently, we introduce a method of measuring the feature importance. Lastly, we summarize the machine learning algorithms used in this work as basis for our machine learning models.

3.4.1. Assessing machine learning model accuracy

Assume we have trained a machine learning model on a data set comprising n_{obs} observations with known input–output pairs. The metrics of assessing the accuracy of a machine learning model depend on whether we use it for a regression or classification problem.

For regression problems, the mean squared error (MSE)

$$\mathrm{MSE} = \frac{1}{n_{\mathrm{obs}}} \sum_{i=1}^{n_{\mathrm{obs}}} \left(Y_i - \tilde{f}(\boldsymbol{X}_i) \right)^2 \tag{3.42}$$

is the most-commonly used measure for the accuracy of a machine learning model [53]. $\boldsymbol{X}_i$ and Y_i are the input and true output of the i^{th} observation of the data set, respectively. A lower MSE means that the machine learning model is more accurate in predicting the true Y_i for a given $\boldsymbol{X}$. At this point we determined the MSE for the training set. A low MSE in this case only tells us that the machine learning model is accurate with data it has already seen. However, in most cases we are interested in generating a machine learning model that is able to generalize well, i.e., that is able to achieve a high accuracy on observations that it was not trained with as well. Such a set of observations that is used to test the generalizability is called a test set.

A commonly used technique for validating a machine learning model is cross-validation. Assume we have a set of n_{obs} observations with known input–output pairs. Like before, we want to train a machine learning model to map inputs to their corresponding outputs as accurately as possible in a generalized manner, i.e., if we obtain new observations, we want the accuracy on them to also be high. To this end, we split the available set of observations into two disjunct sets. One is the training set that we use to train the machine learning model. The other one is the test set that we then use to test how well the machine learning model generalizes. Several options for this splitting exist, and often several split, train and test routines are conducted to determine the average accuracy of the machine learning model.

Among these cross-validation methods is the leave-one-out cross-validation (LOOCV). For this method as many splits are done as there are observations, i.e., n_{obs} splits. Each time, one of the observations is part of the test set and all other observations make up the training set. For the LOOCV estimate of the MSE follows

$$\text{MSE}_{\text{LOOCV}} = \frac{1}{n_{\text{obs}}} \sum_{i=1}^{n_{\text{obs}}} \text{MSE}_{i}, \tag{3.43}$$

where MSE_i denotes the MSE obtained when the i^{th} observation was the test set. An advantage of this approach is its consistency. The resulting estimates for the accuracy of the machine learning model are deterministic. Furthermore, because we use as many observations as possible for training the machine learning model, we get a good estimate for the accuracy we would achieve if we had used all observations. But, if n_{obs} is large, LOOCV becomes computationally very expensive.

An alternative method is k-fold cross-validation (kCV), during which k splits are done, based on which train and test sequences are then performed as follows: the entire set of observations is divided into k subsets. In each iteration, one of them is the test set and the others make up the training set. Thus, the MSE estimate of the kCV is

$$\text{MSE}_{\text{kCV}} = \frac{1}{k} \sum_{i=1}^{k} \text{MSE}_{i}, \tag{3.44}$$

where MSE_i denotes the MSE obtained when the i^{th} subset was the test set. Most commonly, five or ten folds are used [53]. One advantage over LOOCV is that fewer split, train and test sequences are performed and therefore computational resources are saved. Another advantage is related to the so-called "bias-variance trade-off." Outlining the concepts behind this advantage is

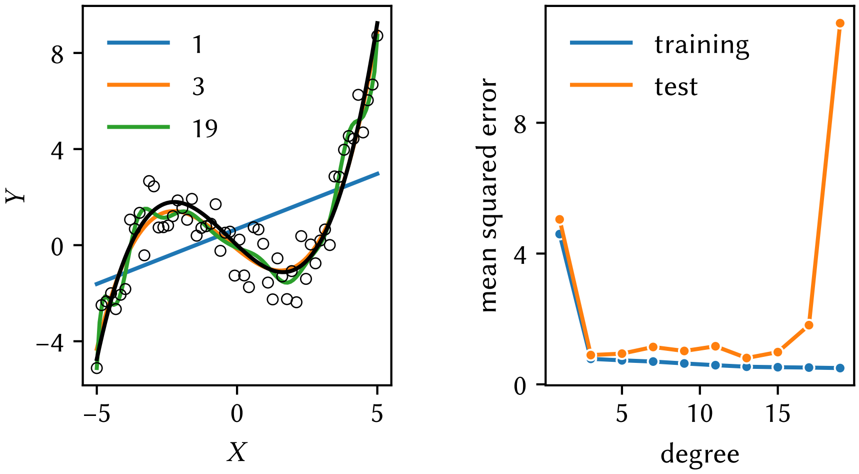


Figure 3.4.: On the left, observations generated from the black line are shown alongside polynomials of different degrees that are fit to the them. On the right, the average mean squared error of the training set and the test set obtained from a 5-fold cross-validation are shown for different degrees of the polynomials.

outside the scope of this work and the interested reader is kindly referred to [53].

In Figure 3.4, we show how the accuracy of different parametrizations of a machine learning models can be determined. In this regression problem, the goal was to train a machine learning model using the observations shown as circles on the left side of Figure 3.4. These circles were generated by adding uniform noise to values obtained from the black curve. This machine learning model has the degree of the polynomial it fits to the data as parameter. The degree of the polynomial represents the flexibility of the machine learning model to approximate a function that accurately maps the inputs of the observations in the training set to their respective outputs. We can see that a high polynomial degree leads to worse approximation of the original function in the form of small oscillations, as it fits the noise in addition to the underlying function.

Performing a 5-fold cross-validation and computing the MSE via equation (3.44) for both the training set and the test set resulted in the values shown on the right side. The MSE of both training set and test set are relatively large for a degree of one. As the degree of the polynomial increases, the MSE of the training set decreases steadily. For the test set, the MSE forms a U-shape.

These trends of the MSE for the training set and the test set with increasing flexibility of the machine learning model are independent of the machine learning model and the data [53]. What changes is the degree of flexibility for which the MSE of the test set is smallest. As this is the configuration we desire most, a lot of effort is put into finding the right degree of flexibility for a given combination of machine learning model and data. In the previous example, it would be determining the degree of the polynomial that results in the smallest error. This process is called hyperparameter optimization.[1]

Why does the MSE of the test set take on a U-shape over the flexibility of the machine learning model? Two competing properties are responsible for this, which are revealed when we decompose the expected value of the MSE for a given input $\boldsymbol{X}_0$, which yields [53]

$$\mathrm{E}\left[\left(Y_0 - \tilde{f}(\boldsymbol{X}_0)\right)^2\right] = \mathrm{Var}\left[\tilde{f}(\boldsymbol{X}_0)\right] + \mathrm{Bias}\left[\tilde{f}(\boldsymbol{X}_0)\right]^2 + \mathrm{Var}\left[\epsilon\right], \tag{3.45}$$

where each term is a nonnegative contribution to the expected value of the MSE, and the last term was discussed before. The first two terms denote the variance and the bias of a machine learning model, and ideally they are both small.

Variance refers to the sensitivity of the machine learning model with respect to the training set. It is large when $\tilde{f}$ changes easily when a different set of observations is used as training set. More flexible machine learning models generally exhibit more variance [53]. If a machine learning model exhibits a large error due to large variance it is said to overfit the data.

Bias refers to the machine learning model being too simple to model the relationship between $\boldsymbol{X}$ and Y for complex real-world problems. For example, regardless of how many observations we used, fitting a line to data whose input–output relationship is nonlinear would not lead to a good approximation. Thus, an inflexible machine learning model results in high bias. If a machine learning model exhibits a large error due to large bias it is said to underfit the data.

It is easy to come up with machine learning models that exhibit either low variance or low bias. On the one hand, if it predicted a constant output irrespective of the training set, it would have low variance but high bias. On the other hand, if it predicted the actual output of the training set for each of its inputs and interpolated between them, it would have low bias but high variance. The challenge is to find a machine learning model that is flexible enough to approximate the underlying relationship of the data, but is not too flexible to additionally approximate fluctuations originating from too small

---

[1] The name hyperparameter stems from the fact that these are specified before the training of the machine learning model, during which its “regular” parameters are adapted to fit the data.

training sets. This interplay of variance and bias of a machine learning model expressed in equation (3.45) is called the bias-variance trade-off.

So far, we have introduced all concepts in the context of regression problems, because those are more accessible. For classification problems, most of the above statements are true as well. The main difference to a regression problem is that the output Y is qualitative, i.e., takes on discrete, non-numerical values that are sometimes called classes. Because of this, machine learning models used for classification problems are also called classifiers. Instead of using the MSE for accuracy assessment, we use the error rate

$$\textsc{er} = \frac{1}{n_{\text{obs}}} \sum_{i=1}^{n_{\text{obs}}} I\big(Y_i \neq \tilde{f}(\boldsymbol{X}_i)\big), \tag{3.46}$$

where I is an indicator variable that equals 1 if its argument is true and 0 if it is false. As the MSE, the error rate may be computed for the training set or the test set, the latter of which should be as small as possible. Although the error rate is the most popular measure for the accuracy of a classifier, we prefer to use the accuracy score

$$\textsc{as} = 1 - \textsc{er}. \tag{3.47}$$

in this work.

When we perform kCV for classification problems, we apply a variant called stratified k-fold cross-validation (SkCV). During the split section of the cross-validation, we make sure that the relative frequency of each class in the training sets and test sets is the same as in the entire data set we split. This way we ensure that a sufficient amount of observations of each class is used when we train the machine learning model.

Now that we are able to assess the predictive power of a machine learning model, the next step for us is to infer as best we can *how* it uses the information of the inputs to predict accurate output. To this end, we determine the so-called permutation feature importance.

3.4.2. Permutation feature importance

Feature importance is a measure of how much the accuracy of a machine learning model depends on one or several features, i.e., input variables. Several methods for determining the feature importance. Some of them are specific to a particular machine learning algorithm, while others can be applied universally. In this work, we compute the permutation feature importance, which was first introduced by Breiman [54] for the random forests machine learning algorithm. Fisher, Rudin, and Dominici [55] then proposed a variation that works irrespective of the machine learning model.

To determine the permutation feature importance of an input variable, we first train our machine learning model using the training set. Subsequently, we assess the accuracy of this machine learning model using the test set. This results in the original score s_{orig} of our machine learning model. Note that we assume that a higher score means that the machine learning model performs better, i.e., it is the opposite to an error measure. We then permute the values of the input variable among all observations within the test set. Thus, the underlying relationship between this input variable and the output that the machine learning model learned is broken. Subsequently, we asses the accuracy of the machine learning model using the modified test set, which yields the permuted score s_{perm}. The permutation feature importance is then given by

$$\textsc{pfi} = s_{\text{orig}} - s_{\text{perm}}, \tag{3.48}$$

i.e., the decrease in accuracy when using the modified test set instead of the original test set. A higher permutation feature importance for a particular input variable means that the machine learning model has learned during training to rely more on it to make accurate predictions compared to other input variables with lower permutation feature importance. This allows us to infer the features of the observations that are the most important in the relationship from inputs to their respective outputs.

Besides determining the permutation feature importance for a single input variable, we can also determine the collective permutation feature importance for a set of input variable. Instead of permuting the values of a single input variable among all observations within the test set, we permute the values of each input variables in this set simultaneously before determining s_{perm}.

3.4.3. Using dislocation densities as inputs

Assume we have a well characterized dislocation microstructure, i.e., we know the spatial configuration, sense and Burgers vector of every dislocation within the microstructure. This enables us to use the D2C method to extract various dislocation densities in arbitrary subdomains within dislocation microstructure. We therefore end up with one value of each dislocation density we consider for each subdomain. A value of a dislocation density within one subdomain might be a scalar or a tensor, i.e., we may have several scalar values that represent one dislocation density value. Thus, when we use dislocation densities as inputs, then each input variable associated with the dislocation densities is the one independent component of a tensor representing one dislocation density in one subdomain. So we can associate each input variable with

- a subdomain, and hence a position in space,
- a dislocation density, and
- a tensor index.

When this information is combined with the permutation feature importance, we may, e.g., determine the feature importance of a specific dislocation density by permuting all input variables associated with this dislocation density simultaneously. We may also determine the location of important dislocation microstructure characteristics by permuting all input variables of a subdomains for the permutation feature importance.

It is clear that we are extremely flexible with this approach. We can use arbitrary spatial discretizations and dislocation densities. The exact combination that performs best might well depend on the problem we are trying to solve, and later on we will try to provide recommendations based on the studies we conduct. But for now, this freedom comes with some implications for suitable machine learning algorithms, especially when we keep possible future applications on dislocation microstructure databases in mind.

Firstly, the machine learning algorithm has to be able to work well with high dimensionality of the input space. The number of input variables for the same set of dislocation density increases linearly with the number of subvolumes n_{Ω_sub}. For three-dimensional problems it thus increases cubically with the spatial resolution. Furthermore, for each additional dislocation density alignment tensors with order n we take into account, the number of input variables increases by $\binom{n+2}{n} n_{\Omega_\text{sub}}$, where

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{n(n-1)(n-2)\cdots\big(n-(k-1)\big)}{k(k-1)(k-2)\cdots 1} \tag{3.49}$$

is the binomial coefficient. The number of independent components of the dislocation density alignment tensors, i.e., the number of input variables per subdomain, for orders up to 5 are shown in Figure 3.5. Thus, if all independent components of the dislocation density alignment tensors up to the 3^rd-order for a discretization of 5 × 5 × 5 subdomains are taken into account, each observation is described by 2500 input variables. Hence, the chosen machine learning algorithm must work well with a lot of input space dimensions.

Secondly, the machine learning algorithm ideally supports online learning, i.e., it is able to be trained incrementally. As the number of input variables

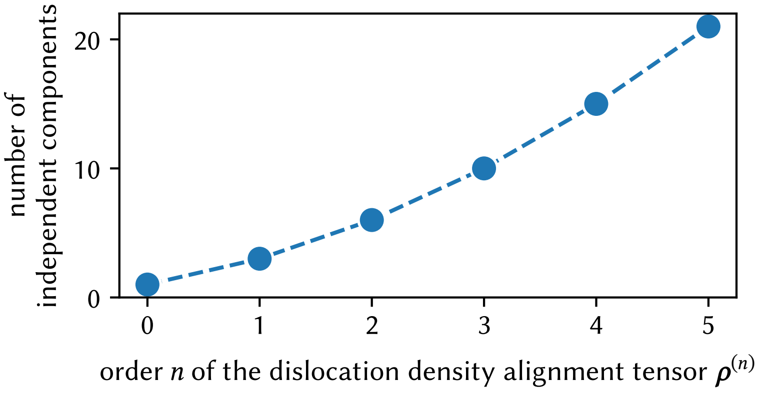


Figure 3.5.: Number of independent components of a dislocation density alignment tensor as function of its order n. The dashed line is shown to guide the eye.

may be very large, the number of observations required to avoid overfitting is also very high. These two factors lead to requiring large amounts of data that might exceed the memory that is available on most workstations. If online learning is possible, we can train out machine learning model incrementally by using one chunk of data at a time, thus avoiding loading all data into memory at once.

Based on these criteria, we have chosen two machine learning algorithms and an additional benchmark machine learning algorithm that does not support online learning, which we base our machine learning models on. These machine learning algorithms are Gaussian naïve Bayes, stochastic gradient descent, and histogram-based gradient boosting machine, respectively. In the following, we summarize these machine learning algorithms and their underlying concepts. Implementations used throughout the present work are that of scikit-learn version 0.22 [56].

3.4.4. Gaussian naïve Bayes

The Gaussian naïve Bayes (GNB) machine learning algorithm is based on Bayes' theorem

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}, \tag{3.50}$$

where A and B are events, $P(\bullet)$ denotes to probability that an event $\bullet$ occurs, and $P(\bullet \mid \circ)$ denotes the conditional probability that an event $\bullet$ occurs given that $\circ$ is true.

Consider a classification problem with n possible outputs Y^j with $j \in \{1, \dots, n\}$. To assign a given observation i with input $\boldsymbol{X}_i$ to its output Y_i, we compute the conditional probability that the corresponding output is Y^j given that the input is $\boldsymbol{X}_i$

$$P(Y^j \mid \boldsymbol{X}_i) = \frac{P(\boldsymbol{X}_i \mid Y^j)P(Y^j)}{P(\boldsymbol{X}_i)} \tag{3.51}$$

for all n possible outputs. Subsequently, the observation is assigned to the output Y^j with the highest probability. As we only compare the probabilities $P(Y^j \mid \boldsymbol{X}_i)$, and $P(\boldsymbol{X}_i)$ is constant for this comparison, we may ignore this contribution. The expressions for the remaining probabilities $P(\boldsymbol{X}_i \mid Y^j)$ and $P(Y^j)$ are determined with the training set comprising n_{obs} observations with known input–output pairs.

The expression for $P(Y^j)$ is the relative frequency with which the output Y^j is observed within the training set

$$P(Y^j) = \frac{1}{n_{\text{obs}}} \sum_{l=1}^{n_{\text{obs}}} I(Y_l = Y^j). \tag{3.52}$$

In contrast, $P(\boldsymbol{X}_i \mid Y^j)$ can not be computed easily as is. The problem is that the input variables X_k with $k \in \{1, \dots, n_{\text{dim}}^{\boldsymbol{X}}\}$ may be conditionally dependent. This means that the probabilities of occurrence for certain values of one or more input variables is affected by the values of other input variables for a given output. If we make the assumption that they are conditionally independent, we may express it as [57]

$$P(\boldsymbol{X}_i \mid Y^j) = \prod_{k=1}^{n_{\text{dim}}^{X}} P(X_{ik} \mid Y^j) \tag{3.53}$$

We then use the probability density function of the univariate Gaussian distribution to model each probability $P(X_{ik} \mid Y^j)$. For this, we compute the arithmetic mean μ_k^j and the standard deviation σ_k^j of the kth input variable for all observations of the training set whose output is Y^j. Thus, we get

$$P(X_{ik} \mid Y^j) = \frac{1}{\sqrt{2\pi(\sigma_k^j)^2}} \exp\left(-\frac{(X_{ik} - \mu_k^j)^2}{2(\sigma_k^j)^2}\right). \tag{3.54}$$

Assembling these expressions for our classification problem then allows us to assign the input $\boldsymbol{X}_i$ to its most likely output Y_i via

$$Y_i = \underset{Y^j, j \in \{1,\dots,n\}}{\arg\max}\ P(Y^j) \prod_{k=1}^{n_{\text{dim}}^{X}} P(X_{ik} \mid Y^j), \tag{3.55}$$

where $\arg\max_\circ \bullet(\circ)$ denotes the arguments of the maximum.

Note that the assumption of conditional independence is likely violated throughout this work. Let us consider we use the components of the 0^{th}- and 2^{nd}-order dislocation density alignment tensors as input for our machine learning models. From equation (2.33) we know that the main diagonal components of $\boldsymbol{\rho}^{(2)}$ add up to $\rho^{(0)}$. The likelihood that these components of our input are conditionally independent is nil. However, Hand and Yu [57] have examined that even though this assumption is violated in a lot of real world data sets, it "seems often to perform surprisingly well," which is in line with our observation, compare sections 6.2.2 and 6.3.2.

3.4.5. Stochastic gradient descent

Let us consider a classification problem with two possible outputs $Y \in \{-1, +1\}$. Our goal is to find a linear function

$$f(\boldsymbol{X}) = \boldsymbol{\beta} \cdot \boldsymbol{X} + \beta_0 \tag{3.56}$$

that best predicts the correct output via

$$Y = \operatorname{sign}\big(f(\boldsymbol{X})\big), \tag{3.57}$$

where $\operatorname{sign} \bullet$ is the sign function. We then consider a loss function $\mathrm{L}\big(f(\boldsymbol{X}), Y\big)$ that allows us to measure the cost of inaccurately predicting the output. Given n_{obs} observations, the error can then be expressed via

$$\mathrm{E}(\boldsymbol{\beta}, \beta_0) = \frac{1}{n_{\text{obs}}} \sum_{i=1}^{n_{\text{obs}}} \mathrm{L}\big(f(\boldsymbol{X}_i), Y_i\big) + \alpha R(\boldsymbol{\beta}), \tag{3.58}$$

where R is a regularization term whose importance is controlled via the weight $\alpha >= 0$. The introduction of the regularization term allows penalization of undesired parametrization of f, e.g., to avoid overfitting.

Our aim is to minimize the error given in equation (3.58). To this end, the stochastic gradient descent (SGD) algorithm is a popular choice. It is a simplification of the regular gradient descent algorithm

$$\boldsymbol{\beta}_{j+1} = \boldsymbol{\beta}_j - \eta \nabla_{\boldsymbol{\beta}} \mathrm{E}(\boldsymbol{\beta}, \beta_0), \tag{3.59}$$

which iteratively finds a better value for $\boldsymbol{\beta}$ by updating its previous value with the weighted gradient of the function that is to be optimized, where the weight η is the so-called learning rate. In this case, all observations are taken into account for the computation of the gradient. The SGD, on the other hand, computes an estimate of this gradient based on a single randomly picked observation j, yielding

$$\boldsymbol{\beta}_{j+1} = \boldsymbol{\beta}_j - \eta_j \nabla_{\boldsymbol{\beta}} \mathrm{E}_j(\boldsymbol{\beta}, \beta_0), \tag{3.60}$$

where the η learning rate may be adapted for each iteration. This enables online learning.

When we choose the so-called hinge loss

$$\mathrm{L}_{\text{hinge}}\big(f(\boldsymbol{X}_i), Y_i\big) = \max\big(0, 1 - Y_i f(\boldsymbol{X}_i)\big) \tag{3.61}$$

as loss function and the square of the Euclidean norm

$$R(\boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{\beta}\|^2 \tag{3.62}$$

as regularization term in equation (3.58), it becomes equivalent to the so-called support vector classifier [58]. As we use these terms in this work and the support vector classifier allows for an easier intuitive understanding of the inner workings of the machine learning algorithm, we outline it in more detail in the following.

3.4.5.1. Maximum-margin classifier

Under the constraint, that the combination of inputs and outputs allows us to define a hyperplane

$$f(\boldsymbol{X}) = \beta_0 + \boldsymbol{\beta} \cdot \boldsymbol{X} = 0 \tag{3.63}$$

that explicitly separates the observations according to their outputs, this hyperplane is called the *separating hyperplane*. The signed distance of the input $\boldsymbol{X}_i$ of an observation i to this hyperplane is then given via

$$d_i = \frac{1}{\|\mathrm{D}_{\boldsymbol{X}} f(\boldsymbol{X}_i)\|} f(\boldsymbol{X}_i) \tag{3.64}$$

Note, that this separating hyperplane may not be uniquely defined, i.e., an infinite amount of hyperplanes that are able to explicitly separate observations according to their outputs may exist. The aim of the maximum-margin classifier is to find the separating hyperplane that maximizes the margin M, i.e., minimum distance, to each observation. This may be expressed in the optimization problem

$$\begin{aligned} \underset{\beta_0, \boldsymbol{\beta}}{\text{maximize}} \quad & M \\ \text{subject to} \quad & Y_i f(\boldsymbol{X}_i) = Y_i(\beta_0 + \boldsymbol{\beta} \cdot \boldsymbol{X}_i) \geq M, \quad i = 1, \dots, n_{\text{obs}}, \\ & \|\boldsymbol{\beta}\| = 1. \end{aligned} \tag{3.65}$$

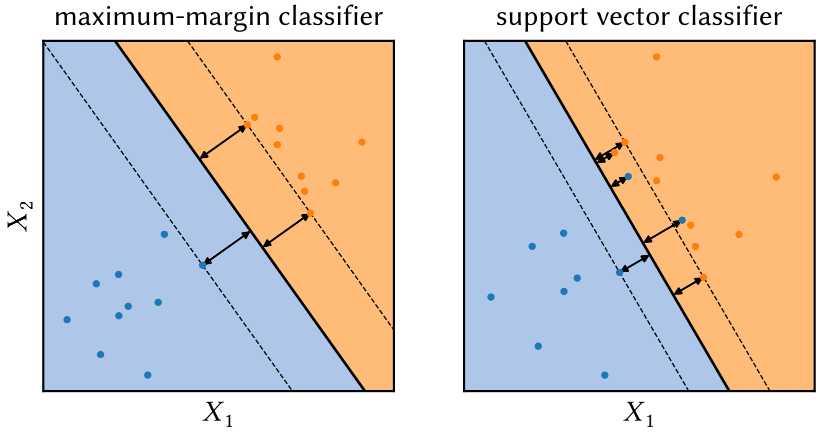


Figure 3.6.: Configuration of a Maximum-margin classifier (left) and a support vector classifier (right) that were fit to the two-dimensional data shown as points. The data contained two different classes, indicated by their color. Shaded areas denote which class an input $\boldsymbol{X}$ would be classified as. The hyperplane is shown as solid line, the margin is indicated by dashed lines. Arrows indicate the distance of the support vectors to the hyperplane.

Once solutions β_0^{opt} and $\boldsymbol{\beta}^{\text{opt}}$ are found, an observation i can be classified via

$$Y_j = \text{sign}\big(f^{\text{opt}}(\boldsymbol{X}_i)\big) = \text{sign}\big(\beta_0^{\text{opt}} + \boldsymbol{\beta}^{\text{opt}} \cdot \boldsymbol{X}_i\big). \tag{3.66}$$

From equation (3.65) follows, that only observations whose input lie on the margin directly influence the configuration of the hyperplane. This can be seen on the left side of Figure 3.6. There, a maximum-margin classifier was fit to data containing two classes. The separating hyperplane is indicated by the solid line, while the margins are indicated by dashed lines. If any of the observations lying on the margin were moved, the configuration of the hyperplane would have to change to follow the optimization problem defined in equation (3.65). Other observations do not influence the configuration of the hyperplane. This small subset of observations therefore “supports” the hyperplane, hence they are called support vectors.

3.4.5.2. Support vector classifier

If the different outputs overlap, we may allow some of their inputs to be within the margin or even on the wrong side of the hyperplane, as seen on the right side of Figure 3.6. To this end, we define so-called slack variables ϵ_i with $i = 1, \dots, n_{\text{obs}}$. Using them, the optimization problem of the maximum-margin classifier defined in equation (3.65) can be adapted for the support vector classifier:

$$\begin{aligned} \underset{\beta_0,\boldsymbol{\beta},\epsilon_1,\ldots,\epsilon_{n_{\text{obs}}}}{\text{maximize}} \quad & M \\ \text{subject to} \quad & Y_i f(\boldsymbol{X}_i) = Y_i(\beta_0 + \boldsymbol{\beta}\cdot\boldsymbol{X}_i) \geq M(1-\epsilon_i), \quad i = 1,\ldots,n_{\text{obs}}, \\ & \|\boldsymbol{\beta}\| = 1, \\ & \epsilon_i \geq 0, \quad i = 1,\ldots,n_{\text{obs}}, \\ & \sum_{i=1}^{n_{\text{obs}}} \epsilon_i \leq C. \end{aligned} \tag{3.67}$$

If $\epsilon_i = 0$, then the input $\boldsymbol{X}_i$ of observation i is on the correct side of the hyperplane and outside of the margin M. If $0 < \epsilon_i <= 1$, then it is on the correct side of the hyperplane, but within the margin. For $\epsilon_i > 1$, it is on the wrong side of the hyperplane. The observations that either lie directly on the margin, or for which $\epsilon_i > 0$, are known as support vectors, as they are the ones determining the configuration of the hyperplane.

As the sum of all slack variables must not exceed C, we might think of it as our tolerance for observations to not conform to the maximum-margin classifier constraints. If $C = 0$, no violations of the maximum-margin classifier constraints are allowed, in which case the support vector classifier is equivalent to the maximum-margin classifier. For small values of C, only few and small transgressions of the margin are allowed. The margin is usually smaller in this case and fewer observations act as support vectors. Furthermore, the support vector classifier follows the data used to train it more closely than for larger values of C. Typically, this value is evaluated as part of the machine learning model selection, using cross-validation to determine a value that best balances the bias-variance trade-off.

3.4.5.3. Extension to more than two possible outputs

So far, we have only considered problems for which we have two possible outputs $Y \in \{-1, +1\}$. However, in this work we study classification problems with more than two possible outputs. A variety of methods to extend aforementioned machine learning algorithms exist, one of the most popular is “one-versus-all” [53]: Assume we have n possible outputs. We then fit n support vector classifiers for each possible output Y^j with $j = 1,\ldots,n$ by encoding the output Y^j as $+1$ and all others as -1. Thus, we get n functions $f^j(\boldsymbol{X}) = \boldsymbol{\beta}^j \cdot \boldsymbol{X} + \beta_0^j$. Finally, we assign an observation with input $\boldsymbol{X}_i$ the output Y^j to for which $f^j(\boldsymbol{X}_i)$ is largest.

3.4.6. Histogram-based gradient boosting machine

The histogram-based gradient boosting machine (HGBM) is a machine learning algorithm that is inspired by LightGBM [59]. It extends upon the concept of the gradient boosting machine (GBM) machine learning algorithm [60]. Outlining the entire algorithm is outside of the scope of this work, as it is far more complex than GNB and SGD, and is used primarily as a benchmark for the other two machine learning algorithms considered in this work due to its recent gain in popularity and success in online competitions. We therefore limit this section to the fundamental concepts contributing to this machine learning algorithms, i.e., boosting and decision trees, and refer the interested reader to Hastie, Tibshirani, and Friedman [58], Friedman [60], and Ke et al. [59].

3.4.6.1. Gradient boosting machine

Boosting is an approach that builds a well performing “committee” by sequentially adding “weak” learners that learn from the residual of the previous committee. A popular and intuitive boosting algorithm introduced by Freund and Schapire [61] is called “AdaBoost.M1”, which we outline in the following.

Let us consider a classification problem for which we have n_{obs} observations with two possible outputs $Y \in \{-1, +1\}$. Initially, we assign equal weight $w_i^1 = 1/n_{\text{obs}}$ to each observation $i = 1, \dots, n_{\text{obs}}$. The following steps are then repeated for $j = 1, \dots, n_{\text{it}}$. Using these weights, we fit a classifier $f^j(\boldsymbol{X})$ to the training data. Subsequently, we compute the error

$$\mathrm{E}^j = \frac{\sum\limits_{i=1}^{n_{\text{obs}}} w_i^j I\big(Y_i \neq f^j(\boldsymbol{X_i})\big)}{\sum\limits_{i=1}^{n_{\text{obs}}} w_i^j}. \tag{3.68}$$

Based on this error, we then compute a classifier weight

$$\gamma^j = \ln \frac{1 - \mathrm{E}^j}{\mathrm{E}^j}. \tag{3.69}$$

Subsequently, we compute the observation weights for the next iteration

$$w_i^{j+1} = w_i^j \exp\Big(\gamma^j I\big(Y_i \neq f^j(\boldsymbol{X_i})\big)\Big), \quad i = 1, \dots, n_{\text{obs}} \tag{3.70}$$

and fit the next classifier.

Once all n_{it} classifiers were fit, the final classifier is defined by

$$f(\boldsymbol{X}) = \operatorname{sign}\left(\sum_{j=1}^{n_{\text{it}}} \gamma^j f^j(\boldsymbol{X})\right). \tag{3.71}$$

Thus, during each iteration a classifier was trained to correct the mistakes done by previous classifiers.

The GBM takes a similar approach, but instead of fitting a classifier to the weighted inputs of all observations it fits the classifier to the negative gradient of a previously chosen loss function. A common choice for the "weak" learners that is also used for the HGBM are decision trees.

3.4.6.2. Classification trees

Classification trees are decision trees for classification problems. Decision trees divide the n-dimensional input space into n-dimensional rectangles, and fit simple functions to each region. The partitioning is done by recursively choosing one input dimension to split, and a value to split it at. When first building the decision trees, this is done in a greedy way. Within a region that we want to split, we search out the combination of input dimension to split, value to split at, and fitting function parameters that decreases the loss function we use most. However, this does not produce an optimal decision tree. The final size of the tree is a hyperparameter that has to be tuned to not under- or overfit the data.

4. Analysis of a notched micro-cantilever beam experiment

TEM is able to visualization single dislocations [62, 63, 64]. As we depicted in Figure 1.1, we see dislocations as black lines on TEM images. Thus, the information directly provided by TEM are two-dimensional projections of the three-dimensional dislocation structure. It is possible to identify the slip system of these dislocations via TEM. A widely applied technique exploits that a dislocation with Burgers vector $\boldsymbol{b}$ can not be seen in a TEM image if the diffraction vector $\boldsymbol{g}$ is oriented such that $\boldsymbol{b} \cdot \boldsymbol{g} = 0$ [65]. But as this requires reorientation of the specimen, it is not practical for in situ experiments.

Another approach proposed by Zhang et al. [66] works by first measuring the angle of a dislocation line in a TEM image with respect to one of the image axes. This angle is then compared with all possible angles of pure edge and pure screw dislocations within the given material. A severe limitation of this approach is the assumption that all dislocations in a specimen are only of pure edge or pure screw type. Especially in experiments where external load is applied, we likely see dislocations of mixed type.

But even if we were able to apply these methods to determine the slip system, this information is not sufficient to fully characterize a dislocation. As we outlined in section 2.2.1, this also requires the determination of its sense. Aforementioned methods are not able to provide this information.

In the following, we propose a combination of geometrical and analytical arguments based on computational analyses to fully characterize the dislocation microstructure of an in situ notched-micro-cantilever beam experiment. With this approach, the two-dimensional dislocation data is used to reconstruct a fully three-dimensional configuration. First, we summarize the creation of a micro-cantilever beam specimen and the relevant details of the experiments, which were performed by Daniel Kiener, Peter J. Imrich, and Inas Issa. Subsequently, we outline the full characterization of the dislocation microstructure observed during the early stages of deformation.

Figure 4.1.: Depiction of the micro-cantilever beam as seen via TEM.

4.1. The experiment

Samples were prepared by initially shaping a chromium single crystal into a wedge. Using a focused ion beam (FIB) with 30 keV gallium cations, a micro-beam cantilever geometry was fabricated. The unavoidable surface damage of this preparation step was removed by exposing the specimen to a temperature of 900 °C for a duration of 20 min. Finally, a notch with a radius of about 2 nm was introduced into the micro-beam cantilever via a condensed 200 keV electron beam. A depiction of the sample is shown in Figure 4.1. The thickness of the micro-beam cantilever is 187 nm. Before conducting the actual experiments, the crystallographic orientation was determined.

4.1.1. Setup

Using this notched micro-cantilever beam, two experiments were performed in sequence. The indenter used for these in situ experiments was a Hysitron Picoindenter PI-95. Synchronized videos of the experiments were recorded using scanning transmission electron microscopy (STEM) with a frame rate of 30 s^{-1}. During the initial loading-unloading experiment, the indenter was displacement controlled at a constant rate of 2 nm s^{-1}. Upon first dislocation activity, the loading is stopped and the indenter is removed. For the subsequent cyclic-loading experiment, the indenter was loading controlled to cyclically load the micro-beam cantilever with a mean load equal to the load at which first dislocation activity was observed in the initial loading-unloading experiment, and an amplitude that is half of it with a frequency of 0.25 s^{-1}.

4.1.2. Results

The resulting time-displacement and time-indenter-force curves are shown in Figure 4.2. In the initial loading-unloading experiment, the displacement increases linearly with time up to 3.32 µN at 38 s. Subsequently, the displace-

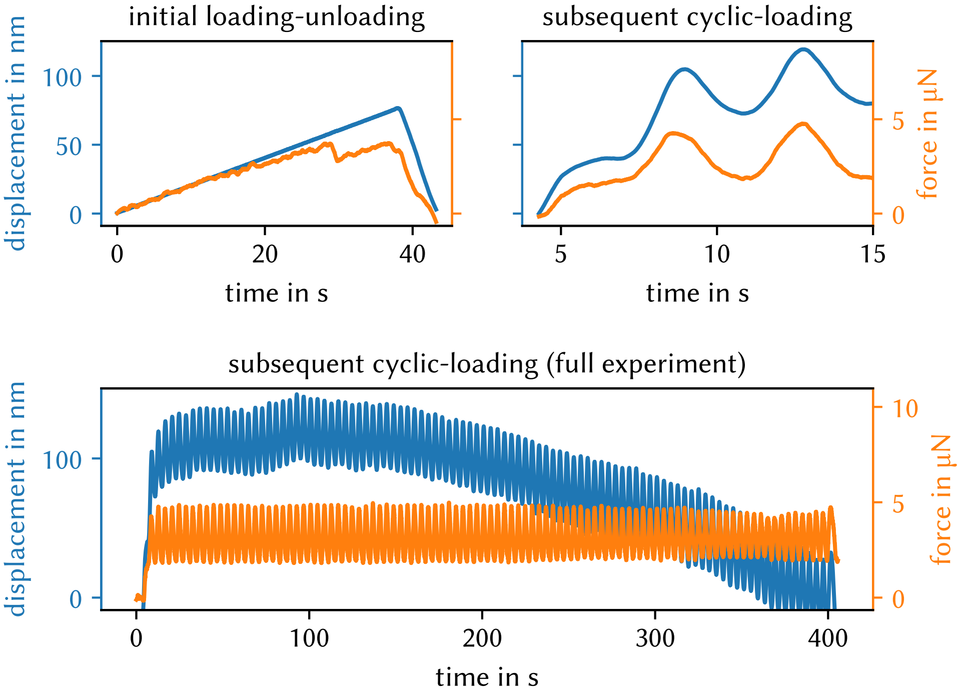


Figure 4.2.: Displacement (blue) and indenter force (orange) over time for the initial loading-unloading (left) and the subsequent cyclic-loading (right) experiments. The slope of the indenter displacement and the indenter force for the first ten seconds of the initial loading-unloading experiment are aligned to highlight deviations from linear behavior.

ment decreases to zero over 5 s. About 29 s into the experiment, the indenter force drops abruptly. In the video, we can observe that the notch opens up at this point. Thereafter, the indenter force continues to increase up to 36.7 s. At this point, the indenter force again drops sharply and after a brief increase decreases with the now decreasing indenter displacement.

In the subsequent cyclic-loading experiment, the indenter comes into contact with the cantilever at about 4 s. Initially, the displacement increases up to 40 nm until the lower indenter force 1.9 µN of the force controlled cyclic-loading scheme is reached. Compared to the initial loading-unloading experiment, a larger displacement is required to reach this force. The indenter is then controlled to cyclically load the cantilever between 1.9 µN to 4.8 µN with a frequency of $0.25\,\mathrm{s}^{-1}$. In the first few cycles, the required displacement to uphold the indenter force acting on the cantilever increases from an average value of 89 nm in the 1^{st} cycle to 114 nm in the 8^{th} cycle at 37 s. After a brief plateau, the average displacement reaches a maximum of 123 nm in the 22^{nd} cycle at 93 s. Subsequently, the average displacement decreases until the end of the experiment.

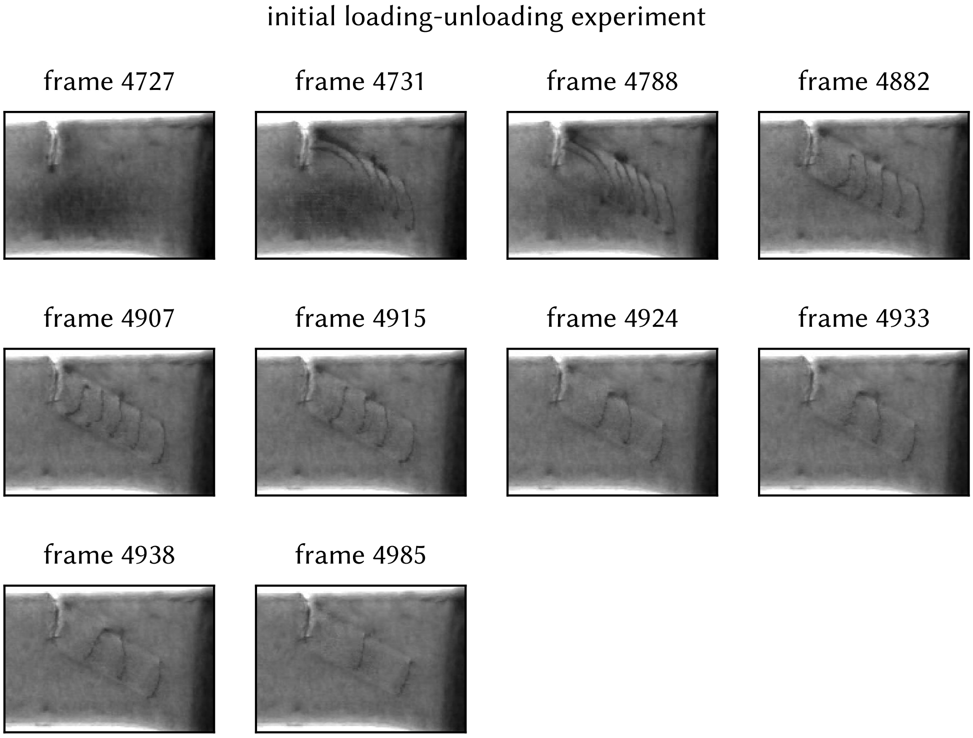
4.2. The analysis

The frames we analyze are shown in Figure 4.3. Within the initial loading-unloading experiment, frame 4727 shows the state of the micro-cantilever beam just before the first dislocations nucleate. After frame 4788, the indenter load is decreased and all except one dislocation slip back towards the notch where they get absorbed by the notch surface. The final configuration of the loading-unloading experiment is shown in frame 1 of the subsequent cyclic-loading experiment. Upon applying a load the dislocation moves slightly until a second dislocation nucleates from the notch just after frame 302.

We take the cubic crystal symmetry of chromium and the resulting anisotropy in the stiffness tensor in all FEM analyses into account. For the elastic properties we used $C_{11} = 339.8\,\mathrm{GPa}$, $C_{12} = 58.6\,\mathrm{GPa}$, and $C_{44} = 99.0\,\mathrm{GPa}$ [67].

4.2.1. Reconstruction of the 3D dislocation microstructure

To uniquely identify the dislocations we analyze in this work, we assign them IDs. The dislocations seen in frame 4788 of the initial loading-unloading experiment are assigned the IDs 1 to 7 from right to left, i.e, in the order they nucleate from the notch surface. Therefore, at frame 4915 of the same

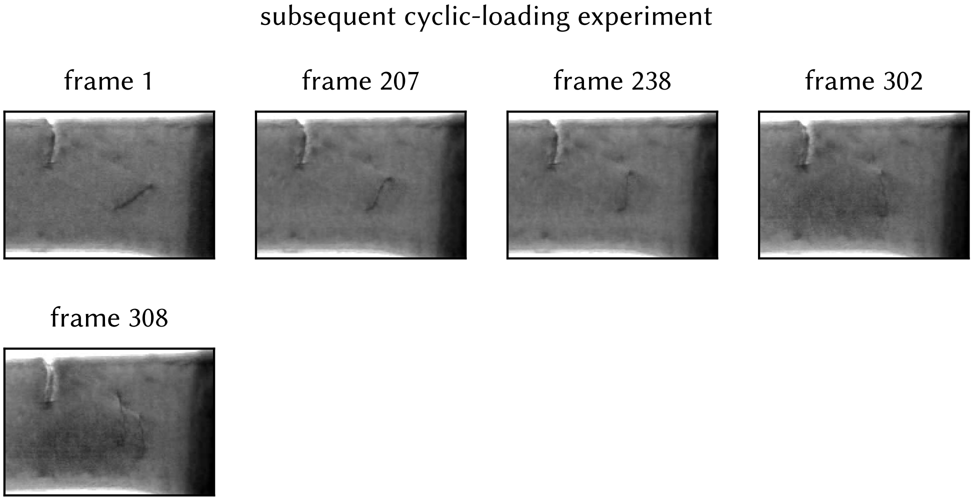


initial loading-unloading experiment

frame 4727 frame 4731 frame 4788 frame 4882

frame 4907 frame 4915 frame 4924 frame 4933

frame 4938 frame 4985



subsequent cyclic-loading experiment

frame 1 frame 207 frame 238 frame 302

frame 308

Figure 4.3.: Frames of the videos recorded during the experiments which are analyzed for the reconstruction of the three-dimensional dislocation microstructure.

experiment we see the dislocations with IDs 4, 3, 2 and 1, from left to right. The additional dislocation seen in frame 308 of the subsequent cyclic-loading gets the ID 8.

As we want to use the FEM to compute the stresses and strains in the three-dimensional configuration, we also need the three-dimensional configuration of the dislocations. Thus, we need to reconstruct the three-dimensional geometry from the two-dimensional frames shown in Figure 4.3. To this end, we first extract the two-dimensional configuration of each dislocation from the TEM images in image coordinates. Along with the dislocations, we also extract the scale bar, which allows us to convert the image coordinates to physical coordinates. In this work, aforementioned extractions are conducted via the open-source software *labelme* [68]. This yields one polygonal chain per dislocation per frame. Given the slip plane $\boldsymbol{n}$ of a slip system and a projection direction $\boldsymbol{d}$, we may project the n_v vertices $\boldsymbol{v}_i$ of a polygonal chain onto the slip plane via

$$\boldsymbol{v}_i' = \boldsymbol{v}_i + \frac{\left(\frac{1}{2}(\boldsymbol{v}_1 + \boldsymbol{v}_{n_\mathrm{v}}) - \boldsymbol{v}_i\right) \cdot \boldsymbol{n}}{\boldsymbol{d} \cdot \boldsymbol{n}}\, \boldsymbol{d}. \tag{4.1}$$

The projection direction coincides with the viewing direction of the TEM images. Conceptually, equation (4.1) yields the intersection point of a line and a plane. The line is defined by 1. the vertex $\boldsymbol{v}_i$ of the two-dimensional configuration, where the missing coordinate takes on an arbitrary value, and 2. the viewing direction $\boldsymbol{d}$. The plane is defined by 1. the midpoint between the initial vertex $\boldsymbol{v}_1$ and the terminal vertex $\boldsymbol{v}_{n_\mathrm{v}}$, and 2. the normal vector $\boldsymbol{n}$ of the slip plane. Their intersection point then is the vertex $\boldsymbol{v}_i'$ of the three-dimensional dislocation configuration that corresponds to the vertex $\boldsymbol{v}_i$ of the two-dimensional one.

However, we do not have any information about the slip systems of the dislocations. In the following, we therefore use a chain of arguments to deduce the slip system and sense of each dislocation, starting with the dislocation with ID 1.

We know the possible slip systems in a BCC material. The first argument is a purely geometrical one: If we project the two-dimensional polygonal chain obtained from the TEM images onto a slip plane, then the extent of the three-dimensional polygonal chains in viewing direction should be the same as the thickness of the micro-cantilever beam, regardless of the frame the dislocation is in. We therefore compute the three-dimensional polygonal chain for all slip systems and frames using equation (4.1). The relative error of the depth of the dislocation with ID 1 with respect to the micro-cantilever beam thickness is shown in Figure 4.4. The slip systems are sorted with respect to

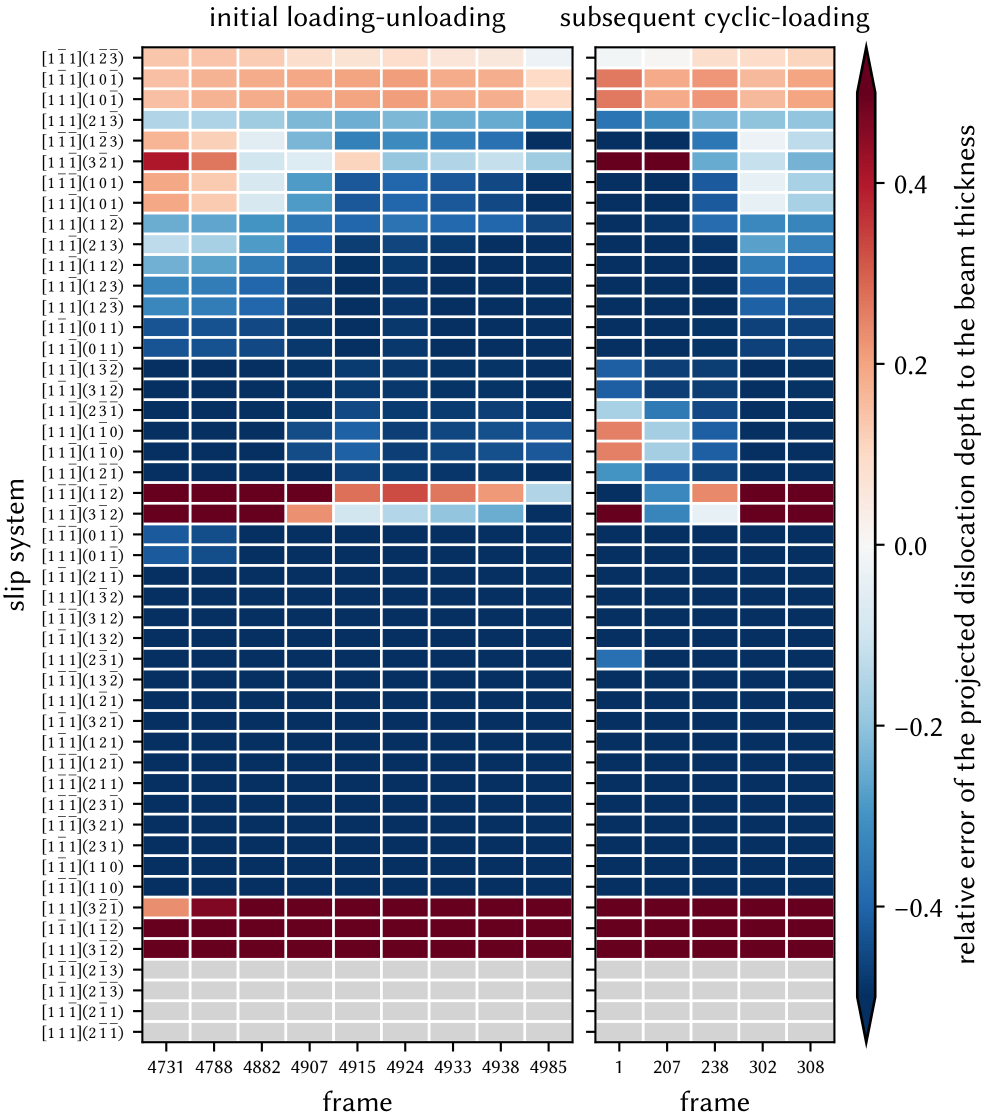


Figure 4.4.: Relative error of the depth of the dislocation line with the ID 1 projected onto each possible slip system of the BCC crystal structure with respect to the actual beam thickness 187 nm for different frames in the initial loading-unloading and subsequent cyclic-loading experiments.

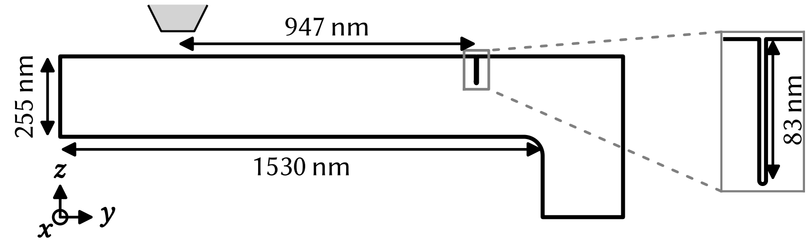


Figure 4.5.: Idealized geometry of the notched micro-cantilever beam used for the FEM analysis. The thickness of the beam is 187 nm. The indenter is depicted by the gray-shaded area. The radius of the notch is 2 nm.

their average error over all frames. We observe that most slip systems can be excluded as the extent of the projected dislocation configuration is either far too small or large. The slip systems shown in the bottom rows have slip planes that are parallel to the viewing direction, so they can also be excluded. We now consider the five slip systems with the least average error, i.e., the top five rows, as remaining candidate slip systems for the dislocation with ID 1.

To further narrow the list of viable slip systems down, we now perform a FEM analysis of the micro-beam cantilever during frame 4727 of the initial loading-unloading experiment, i.e., just before the dislocation with ID 1 nucleated from the notch surface. The argument we want to make is the following: For a dislocation to nucleate, the driving force acting on its slip system due to stresses must be fairly large. Thus, when the resolved shear stress of a slip system is small compared to the resolved shear stresses of other viable slip systems, we may also rule it out.

The idealized geometry we used for the FEM analysis is shown in Figure 4.5. We prescribed zero displacement boundary conditions on the surfaces furthest in positive y-direction and furthest in negative z-direction. To emulate the indenter, we prescribed traction boundary condition equivalent to a force of 3.69 µN. We applied adaptive mesh refinement repeatedly until the quantities of interest exhibit convergence. During the analysis, we also look into a possible simplification that could potentially allow us to bypass a full FEM analysis. As we load the micro-cantilever beam, we expect large compressive stresses in y-direction at the notch tip. We compute the Schmid factors under the assumption that this compressive stress is much larger than other stresses, i.e., by treating it as uniaxial compression.

We show the Schmid factor and resolved shear stress maximum at the notch surface for each possible slip system in the BCC crystal structure just before the nucleation of the first dislocation in the initial loading-unloading experiment in Figure 4.6. The bars indicate the average resolved shear stress maximum of seven FEM analyses performed for uniformly spaced crystal axes alignment errors between −3° to 3° from the experimentally measured 8°. Compared to the difference between slip systems, the possible error resulting from the uncertainty in the experimental determination of the crystal orientation is small. Superscripts in front of the slip systems denote the five slip systems that lead to low relative projected depth errors, compare Figure 4.4. The slip system $[1\,\bar{1}\,1](1\,0\,\bar{1})$ – one of the slip system candidates – exhibits the highest observed resolved shear stress maximum with 5.79 GPa. This is 2.3 % higher than the next highest observed resolved shear stress maximum overall, and 22 % higher than for the next highest slip system candidate for this dislocation, $[1\,\bar{1}\,1](1\,\bar{2}\,\bar{3})$. The Schmid factors as a whole follow the trend of the resolved shear stresses, but their deviations from the resolved shear stresses are very large. This indicates that the Schmid factors assumption is not a viable substitution for the full FEM analysis due the triaxial stress state at the notch tip.

Based on the low relative error of the depth of the dislocation line with respect to the beam thickness and the analysis of the resolved shear stress maxima, we conclude that the most likely slip system of the dislocation with ID 1 is $[1\,\bar{1}\,1](1\,0\,\bar{1})$. However, to fully characterize the dislocation, we also need to determine its sense. To this end, we consider two criteria; 1. the elastic energy of the micro-cantilever beam depending on the sense, and 2. the direction of the Peach–Köhler force the dislocation would experience along the intersection of its slip plane with the notch surface. In Figure 4.7, we show the geometric entities used to define the sense of a dislocation in this study. We define the sense of the dislocation according to the direction with respect to its dislocation slip area and the normal vector thereof. Thus, there are two possible senses; a right-handed and left-handed senses.

We expect the stored elastic energy in the micro-cantilever beam to differ depending on the sense of the dislocation. A sense resulting in lower total elastic energy within the same configuration may be more likely. To this end, we compute the total elastic energies for frames in which only this dislocation is present, i.e., frames 1, 207, 238 and 302 of the cyclic-loading experiment, are computed via the FEM for both senses. As we are mostly interested in which of the two yield less total elastic energy, we determine the relative deviations of the elastic energy for each frame via $\left(E^{\mathrm{el}}(\boldsymbol{\xi}) - \langle E^{\mathrm{el}}\rangle\right)/\langle E^{\mathrm{el}}\rangle$, where $\langle E^{\mathrm{el}}\rangle$ denotes the frame-wise mean. The results are shown in Figure 4.8. We observe that the total elastic energy of the beam is lower for the left-handed sense, regardless of the frame.

The resolved shear stress the dislocation with ID 1 would experience at the surface of the notch in frame 4727 of the initial loading-unloading experiment,

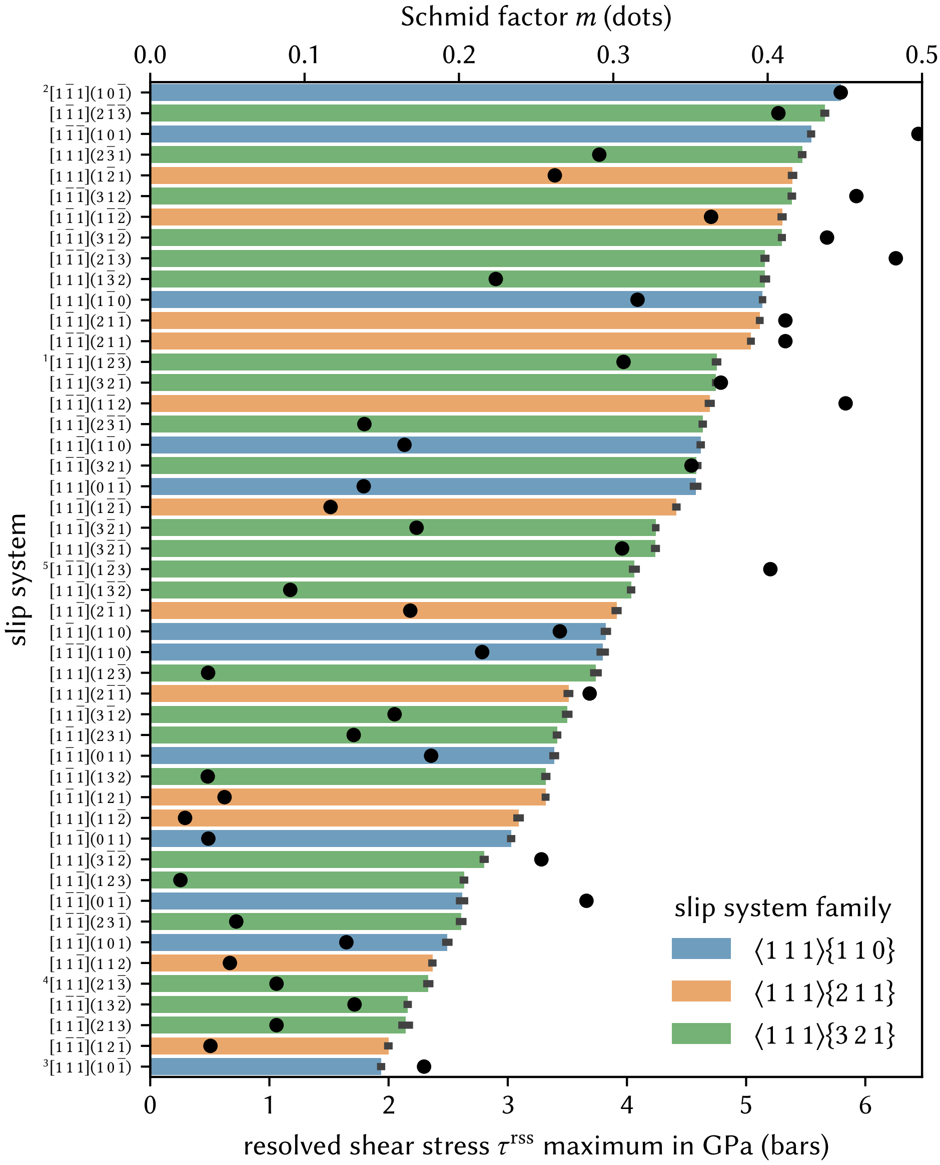


Figure 4.6.: Schmid factor (dots) and resolved shear stress maximum (bars) at the notch surface for each slip system in BCC observed in frame 4727 of the initial loading-unloading experiment, i.e., just before the nucleation of the first dislocation. The Schmid factor of the slip system with the highest observed resolved shear stress maximum was aligned with the latter. Bar colors correspond to the slip system family. Error bars denote the standard deviation of the sample of the resolved shear stress maximum for different misalignment errors. Superscripts before the slip system denote the five slip system who are the most likely candidates according to their low relative error in the projected depth, see Figure 4.4.

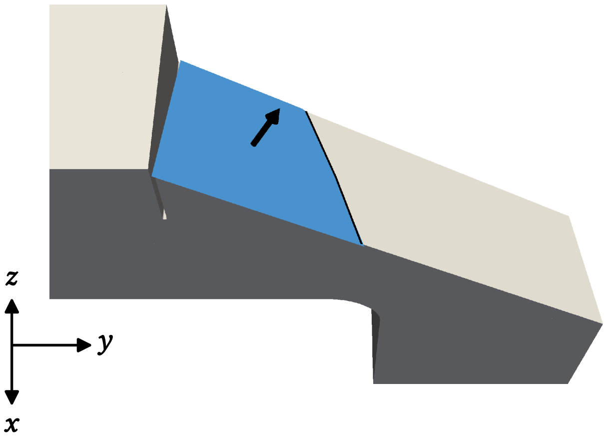


Figure 4.7.: Micro-beam cantilever cut to reveal the spatial configuration of the dislocation with ID 1, shown as black line. Its dislocation slip area is depicted by the blue area, where the black arrow represents its normal vector. Based on the normal vector and the dislocation slip area, we define the sense of the dislocation as either right-handed or left-handed.

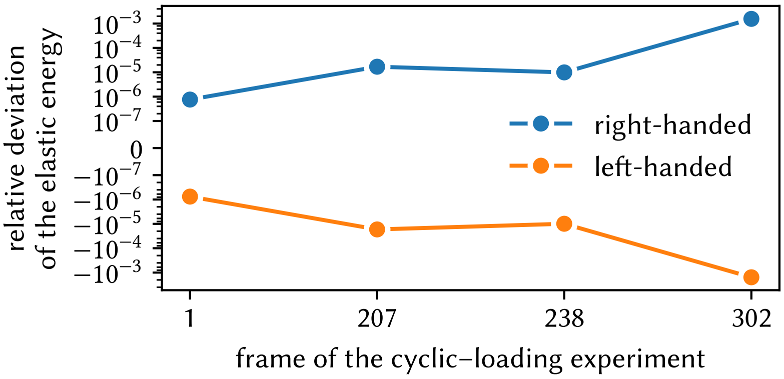


Figure 4.8.: Relative deviation of the total elastic energy for frames that only include the dislocation with ID 1 for different senses.

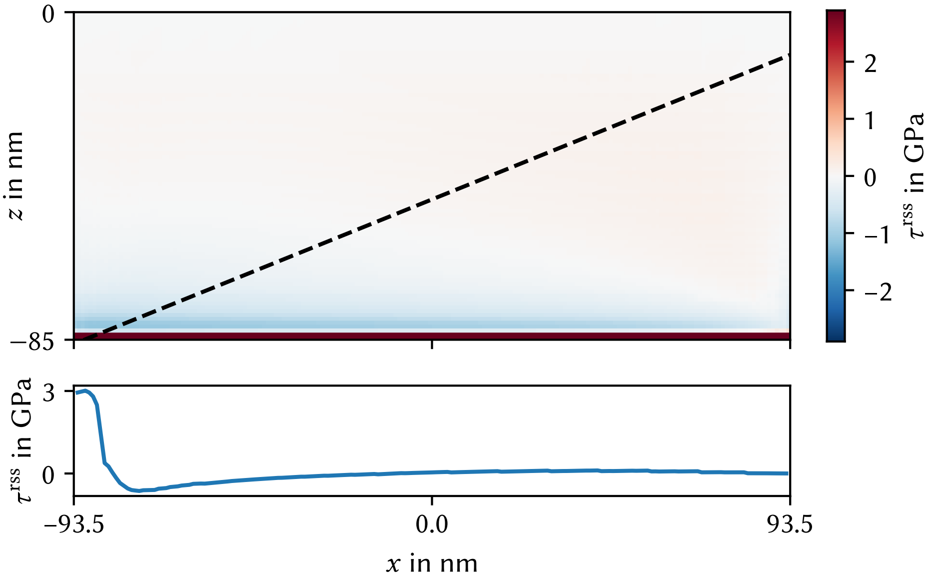


Figure 4.9.: The resolved shear stress τ^{rss} for the slip system $[1\,\bar{1}\,1](1\,0\,\bar{1})$ at the notch surface which dislocations nucleate from is shown at the top for frame 4727 of the loading-unloading experiment. A right-handed sense of the dislocation is assumed, thus a positive τ^{rss} acts to drive the dislocation away from the notch and into the material. The dashed line indicates the sites at which the dislocation with ID 1 could nucleate, i.e., it is the intersection of the notch surface with its slip plane. In the bottom, the resolved shear stress along this line is depicted.

i.e., just before its nucleation, is shown on the bottom side in Figure 4.9. The intersection of the slip plane with the notch surface is shown on the upper side as dashed line. We show the resolved shear stress along this line in the bottom.

The highest observed value of the resolved shear stress of 3.01 GPa is observed at the notch tip and carries a positive sign. With increasing x along the intersection, the value decreases and exhibits a minimum −0.46 GPa. Subsequently, the value trends towards 0 GPa. Thus, if the dislocation had a right-handed sense, it would experience a force driving it away from the notch at the notch tip and into the material.

While the total elastic energy suggests a left-handed sense, these energies are computed for states in which the dislocation had been nucleated and slipped into the material. However, the Peach–Köhler force the dislocation would experience at the notch tip if it had left-handed sense would pull it towards the notch surface instead of into the material. *Therefore, we conclude is that the dislocation with the ID 1 is of the* $[1\,\bar{1}\,1](1\,0\,\bar{1})$ *slip system with a* *right-handed sense.*

Using similar arguments, we can determine the slip system and sense of the dislocation with ID 8, i.e., the second dislocation to be nucleated during the cyclic-loading experiment.

Firstly, we identify the slip system. Similarly to how we identified the slip system for the dislocation with ID 1, we start by comparing the extent of the projected three-dimensional dislocation configuration in viewing direction for every possible slip system with the micro-beam cantilever thickness. We show the relative errors in Figure 4.10. This time, we consider the seven slip systems with the smallest average errors as viable slip system candidates. Note that the slip system assigned to the dislocation with ID 1 is among them.

The resolved shear stress maxima at the notch surface for all 48 possible slip systems are shown in Figure 4.11. Superscripted indices precede the seven candidate slip systems, where the index denotes the placement with respect to least error in the projected extent in viewing direction. The overall highest observed resolved shear stress maximum of 3.57 GPa is from the slip system $[1\,1\,1](2\,\bar{3}\,1)$. Of the seven slip systems candidates, $[1\,\bar{1}\,\bar{1}](1\,0\,1)$ exhibits the highest resolved shear stress maximum with 3.33 GPa. This is about 6 % higher than the one we observe for the slip system $[1\,\bar{1}\,1](1\,0\,\bar{1})$, whose resolved shear stress maximum is 3.14 GPa. The next highest resolved shear stress maximum 2.38 GPa of the slip system slip system $[1\,\bar{1}\,\bar{1}](1\,\bar{2}\,3)$ is about 24 % lower than that.

Due to their low error of the depth of the projected dislocation line and their high resolved shear stress maxima, $[1\,\bar{1}\,\bar{1}](1\,0\,1)$ and $[1\,\bar{1}\,1](1\,0\,\bar{1})$ are viable slip systems so far. We therefore consider on more argument: As we know that the dislocation originates from the notch surface, the slip traces left on the sides of the micro-cantilever beam must outline the path travelled by the dislocation. We determine the slip traces on the micro-cantilever beam surface by intersecting the surfaces with the slip plane. The resulting potential slip traces are shown in Figure 4.12. We can clearly see that the slip system $[1\,\bar{1}\,\bar{1}](1\,0\,1)$ can be ruled out as its slip traces would originate from the bottom of the micro-cantilever beam instead of from the notch surface. Therefore, we conclude that the slip system of dislocation with ID 8 is $[1\,\bar{1}\,1](1\,0\,\bar{1})$.

To determine the sense, we again consider the driving force acting on the dislocation shortly after its nucleation at the notch surface. The resolved shear stress it would experience at the surface of the notch in frame 308 of the subsequent cyclic-loading experiment, i.e., just before its nucleation, is shown on the bottom side in Figure 4.13. The intersection of the slip plane with the notch surface is shown on the upper side as dashed line. We show the resolved shear stress along this line in the bottom.

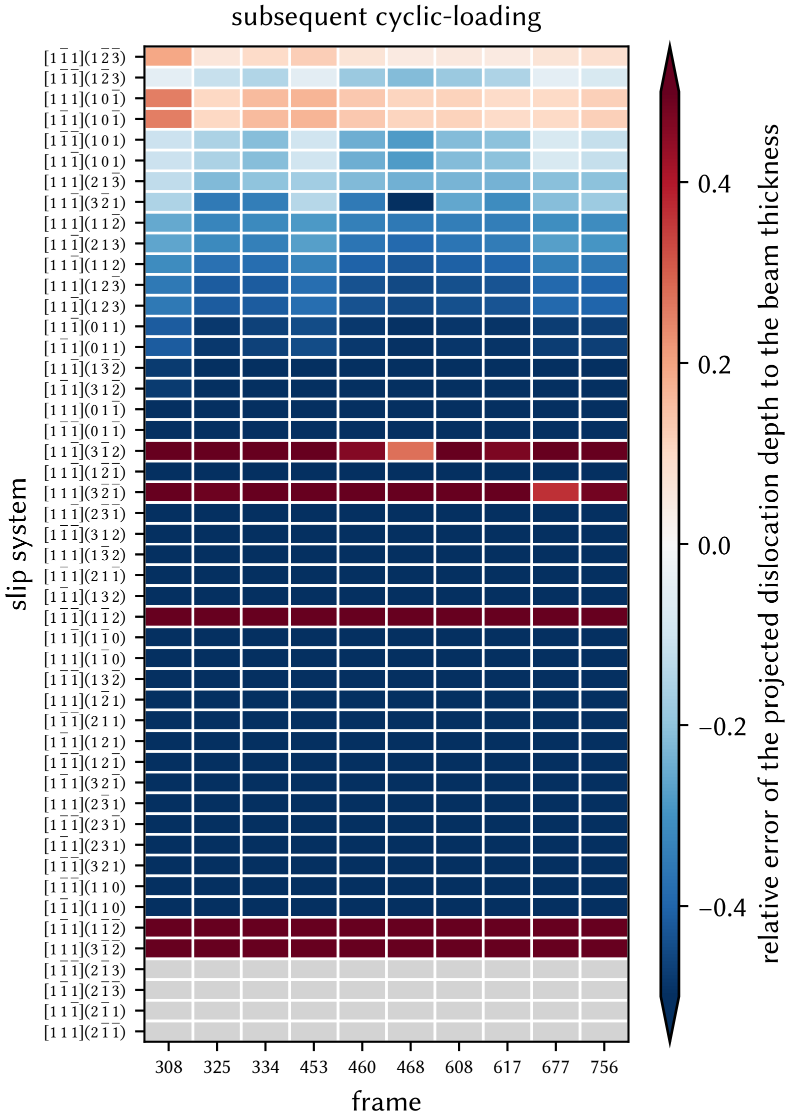


Figure 4.10.: Relative error of the depth of the dislocation line with the ID 8 projected onto each possible slip system of the BCC crystal structure with respect to the actual beam thickness 187 nm for different frames in the cyclic-loading experiment.

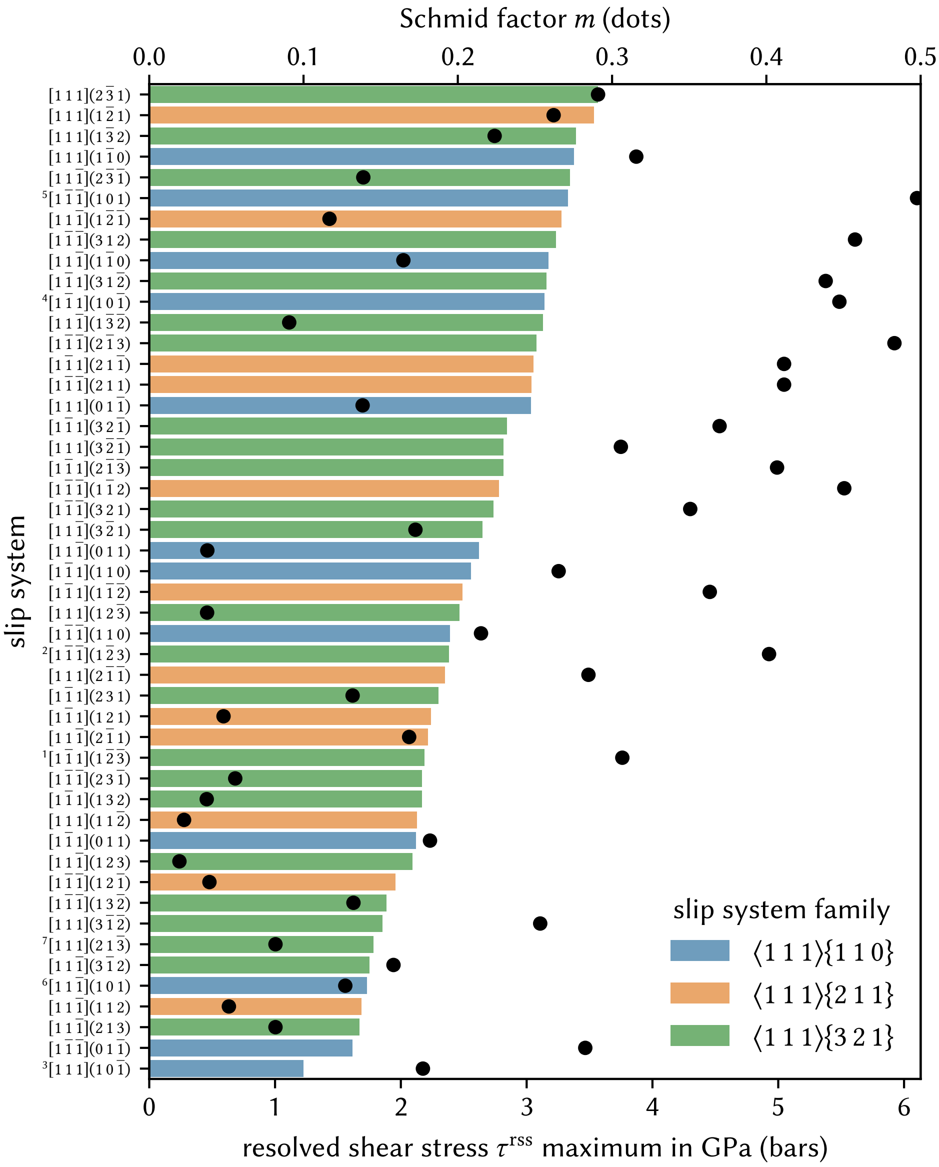


Figure 4.11.: Schmid factor (dots) and resolved shear stress maximum (bars) at the notch surface for each slip system in BCC observed in frame 302 of the cyclic-loading experiment, i.e., just before the nucleation of the second dislocation. The Schmid factor of the slip system with the highest observed resolved shear stress maximum was aligned with the latter. Bar colors correspond to the slip system family. Error bars denote the standard deviation of the sample of the resolved shear stress maximum for different misalignment errors. Superscripts before the slip system denote the seven slip system who are the most likely candidates according to their low relative error in the projected depth, see Figure 4.10.

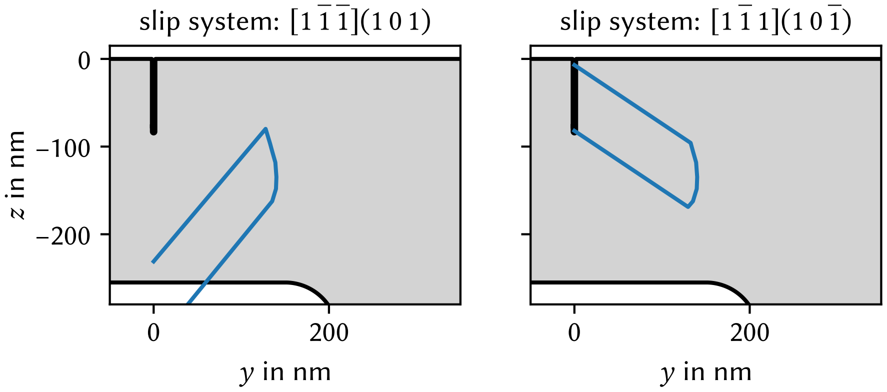


Figure 4.12.: Line and slip traces the dislocation with ID 8 would leave on the surface of the specimen if its slip system were $[1\bar{1}\bar{1}](1\,0\,1)$ or $[1\bar{1}1](1\,0\,\bar{1})$. The configuration used is that of frame 308 of the cyclic-loading experiment, see Figure 4.3.

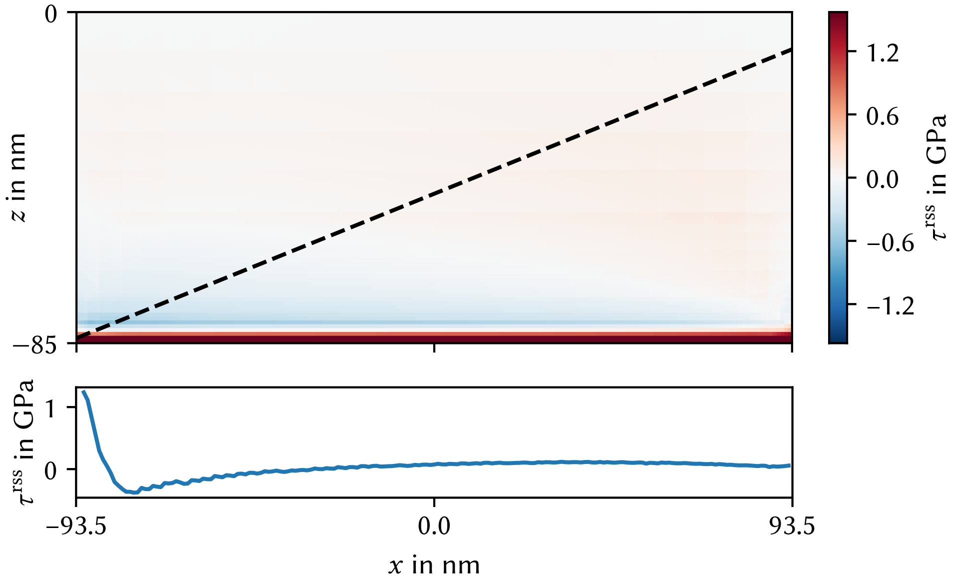


Figure 4.13.: The resolved shear stress τ^{rss} for the slip system $[1\bar{1}1](1\,0\,\bar{1})$ at the notch surface which dislocations nucleate from is shown at the top for frame 302 for the cyclic-loading experiment. A right-handed sense of the dislocation is assumed, thus a positive τ^{rss} acts to drive the dislocation away from the notch and into the material. The dashed line indicates the sites at which the dislocation with ID 8 could nucleate, i.e., it is the intersection of the notch surface with its slip plane. In the bottom, the resolved shear stress along this line is depicted.

Unfortunately, we can not follow the same rigour in our determination of the slip systems and senses of the other dislocations with IDs 2 to 5 seen in the initial loading-unloading experiment. This is because we lack TEM images showing the states just before they nucleate. Thus, an in-depth analysis of the resolved shear stresses before their nucleation is not possible.

The relative errors of the projected dislocation depths with respect to the beam thickness for these dislocations are shown in figures 4.14 to 4.17. In all cases, the slip system $[1\,\bar{1}\,1](1\,0\,\bar{1})$ is among the slip systems with the lowest error.

We have seen in the analysis for the dislocation with ID 8 that the slip system $[1\,\bar{1}\,1](1\,0\,\bar{1})$ exhibits a high resolved shear stresses compared to other slip systems. We may therefore assume that this is also the case for the other dislocation. Similarly, we may also assume these dislocations to have right-handed sense, as the overall character of the resolved shear stress did not change between dislocations with IDs 1 and 8, see figures 4.9 and 4.13, respectively.

4.2.2. Nucleation of dislocations

Based on the results and analysis so far, we may infer how the first dislocations are nucleated from the notch tip. As seen in figures 4.9 and 4.13, the dislocations only exhibit forces pushing it into the material at the tip of the notch. Just above the notch tip, dislocations are actually pulled towards the surface. To get a better idea of the driving forces, the resolved shear stress on the slip plane of the dislocation with ID 1 are shown in Figure 4.18. A positive resolved shear stress acts to expand the dislocation, driving it into the material. A negative one acts to shrink it, driving it towards the notch.

The only way a dislocation that nucleates at the notch surface is able to expand therefore is by doing so at the tip of the notch. Upon the initial expansion, the part of the dislocation in contact with the notch surface experiences forces that act to push it back to the notch tip. If the dislocation shrank close to the surface, its curvature would increase. Acting against this is the line tension, which is “equivalent to a thermodynamic force that acts to reduce dislocation line length.” [31] The front of the dislocation that is being pushed into the material therefore “drags along” the part connected to the notch surface. Eventually, this part reaches the front side of the beam and is pushed into the material. Further into the material, the resolved shear stress decreases and after about $y \approx 150\,\mathrm{nm}$ changes sign. This is similar to the neutral axis of a regular bending beam. The configuration of the dislocation with ID 1 for frame 1of the cyclic-loading experiment is shown as black line. It aligns

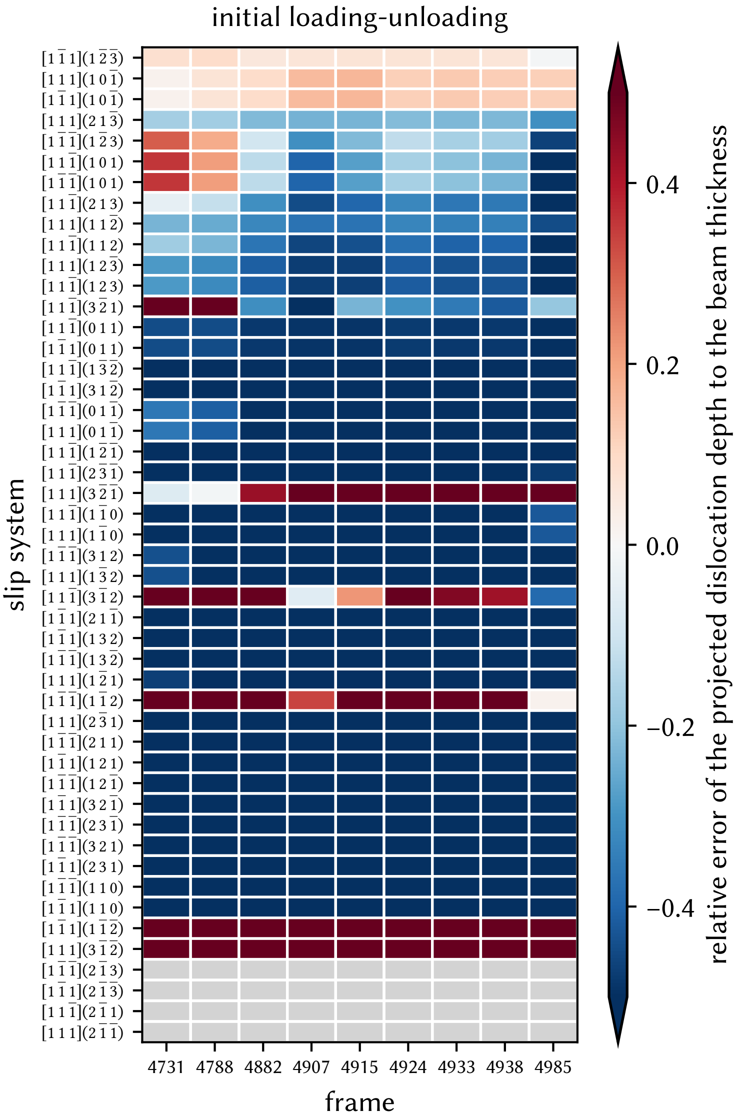


Figure 4.14.: Relative error of the depth of the dislocation line with the ID 2 projected onto each possible slip system of the BCC crystal structure with respect to the actual beam thickness 187 nm for different frames in the initial loading-unloading experiment.

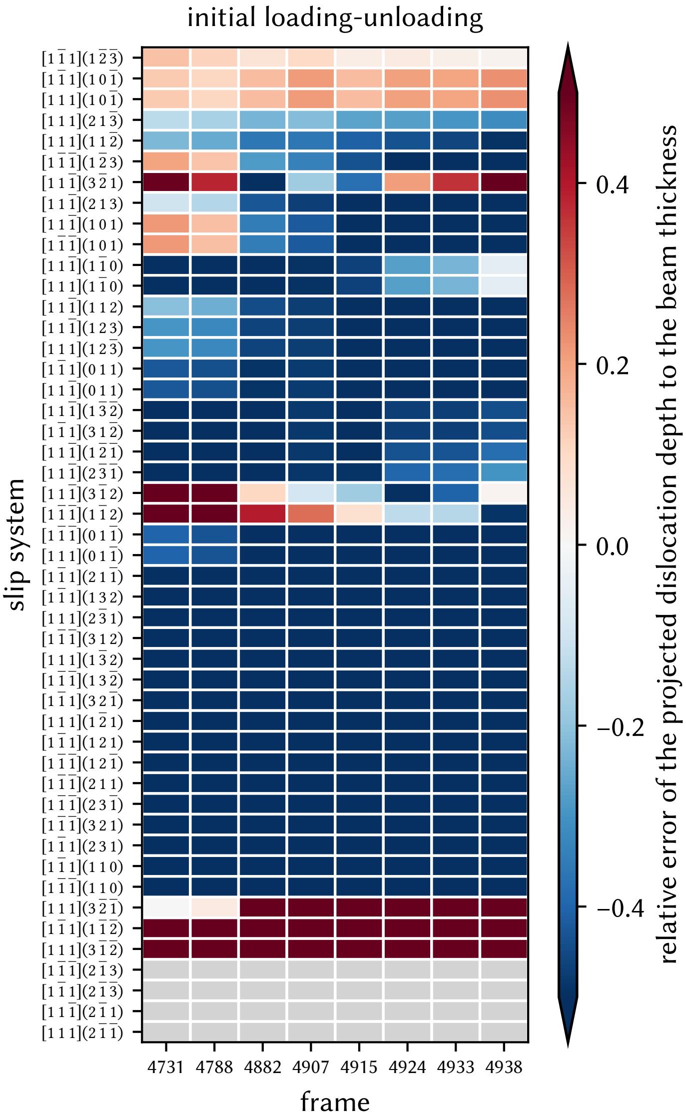


Figure 4.15.: Relative error of the depth of the dislocation line with the ID 3 projected onto each possible slip system of the BCC crystal structure with respect to the actual beam thickness 187 nm for different frames in the initial loading-unloading experiment.

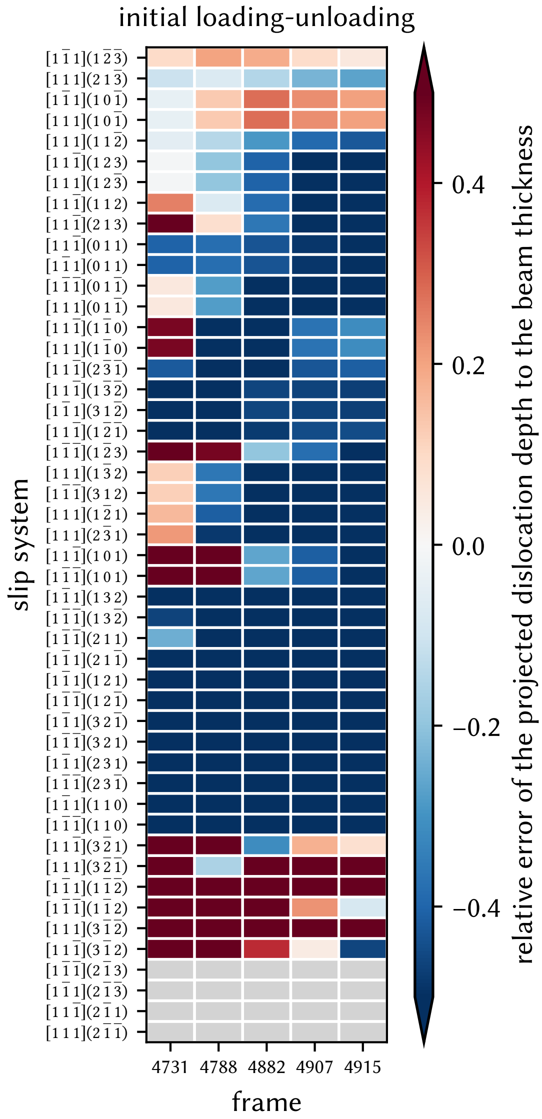


Figure 4.16.: Relative error of the depth of the dislocation line with the ID 4 projected onto each possible slip system of the BCC crystal structure with respect to the actual beam thickness 187 nm for different frames in the initial loading-unloading experiment.

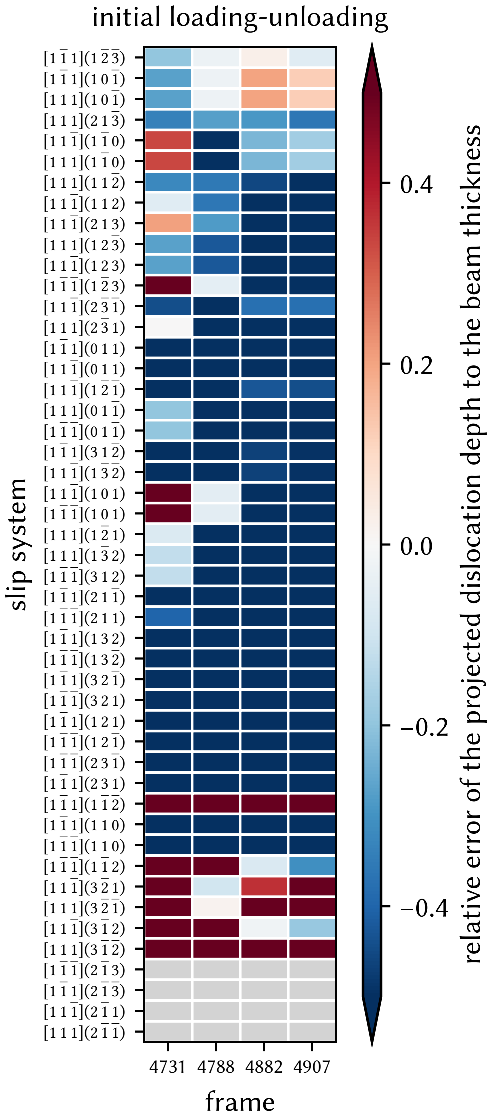


Figure 4.17.: Relative error of the depth of the dislocation line with the ID 5 projected onto each possible slip system of the BCC crystal structure with respect to the actual beam thickness 187 nm for different frames in the initial loading-unloading experiment.

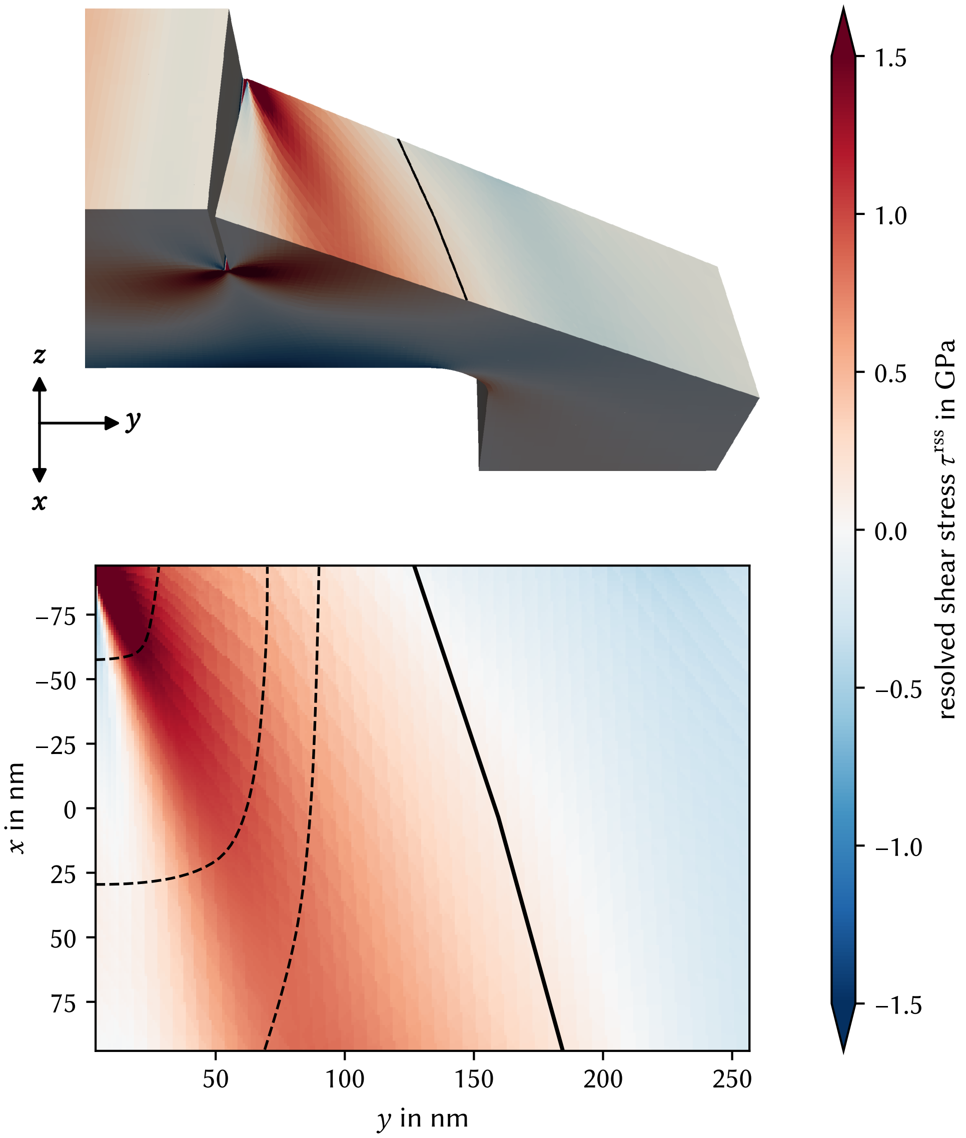


Figure 4.18.: Resolved shear stress of the slip system $[1\,\bar{1}\,1](1\,0\,\bar{1})$ shown for the slip plane of the dislocation with ID 1. At the top, the micro-cantilever beam was cut along the slip plane . At the bottom, the slip plane of this dislocation is viewed along the z-axis. The configuration of the dislocation for frame 1 in the cyclic-loading experiment is shown as black line in both. Dashed lines indicate possible intermediate configurations of the dislocation during its propagation.

well with the neutral axis of this beam. Furthermore, this also explains why it did not get pulled towards and absorbed by the notch surface when the load was decreased in the initial loading-unloading experiment. A possible sequence of configurations this dislocation took on is shown as dashed lines in Figure 4.18.

4.3. Summary and conclusion

In this chapter, we outlined how a notched micro-cantilever beam was produced in a way that minimizes the number of defects that might affect its plastic deformation behavior.

Based on the geometry of the micro-cantilever beam and the recorded images of the dislocation microstructure, we developed an approach that is able to fully characterize each dislocation. To this end, we used arguments based on the thickness of the micro-cantilever beam and stresses computed at the notch surface to narrow down the possible slip systems. In this manner, we were able to fully reconstruct the three-dimensional dislocation microstructure from images recorded via TEM during an in situ experiment. We observed that a simplified approach via computation of the Schmid factor could lead to wrong conclusions. Using the internal stress state of the micro-cantilever beam, we were then able to infer how the first dislocation propagated from the notch tip into the material.

We conclude that the combination of modern microscopy techniques with computational analyses tools is able to greatly enhance the information we can obtain from in situ experiments. The reconstruction of three-dimensional dislocation microstructures is possible based on a combination of geometrical and analytical arguments. Dislocation microstructures reconstructed in this manner may in the future be used to validate other computational approaches like, e.g., DDD.

5. Initial dislocation configurations for discrete dislocation dynamics

There are various ways how the initial dislocation microstructure for discrete dislocation dynamics simulations can be created. One way is by using Frank–Read sources (FR sources), i.e., dislocation segments that additionally have fixed pinning points. They can act as artificial sources that emit dislocations. However, using FR sources is rather unphysical, as the rigid pinning points prohibit otherwise possible ways of motion.

Another popular approach is to use prismatic dislocation dipoles [69]. These dislocation lines are closed but span different slip planes with the same Burgers vectors. The movement of the dislocation is thus constrained similarly to the FR sources, but without the pinning points being completely rigid. In this manner, they act as flexible dislocation sources.

Tang, Schwarz, and Espinosa [70] used a combination of randomly distributed straight and jogged dislocation lines for the initial conditions. Jogs are parts of the dislocation line whose line direction is perpendicular to the slip planes of the dislocation. Thus, they may only move via nonconservative motion which effectively means that jogs act as pinning points. In the stress-strain curves they observed features similar to “experimentally observed staircase stress-strain response.” [70]

Motz et al. [71] generated dislocation microstructures by randomly placing circular dislocation loops inside and outside of the actual simulation box. Subsequently, they allowed the structures to relax, during which, e.g., junctions formed. Thus, the initial dislocation microstructure contained only features that are compatible with the geometric features of dislocations as boundaries of dislocation slip areas. They found that during the evolution of the dislocation microstructures, pinning points formed naturally as a result of cross-slips and other dislocation reactions.

Zhou, Biner, and LeSar [72] generated dislocation microstructures by first setting up a 3 µm × 3 µm × 3 µm simulation box with periodic boundary conditions and filled with several FR sources equal to a total dislocation density of $2 \cdot 10^{12}\,\mathrm{m}^{-2}$ to represent bulk material. After prestraining the structure up to a plastic strain of 0.1 %, it was allowed to relaxed. Subsequently, cylindrical samples were "cut" from this and used for their simulations. They therefore contained combinations of "FR sources, jogged dislocations, surface dislocations and spiral (single-armed) sources." [72] With this initial dislocation microstructure, they found good agreement between their simulations and experiments in terms of stress-strain curves.

Both, Motz et al. [71] and Zhou, Biner, and LeSar [72] also studied the effect of considering cross-slip during the simulations on the stress-strain curves and the total dislocation density of the whole domain. They found that the consideration of cross-slip results in lower stresses for the same magnitude of strain and a higher total dislocation density.

In all of the aforementioned work, the quality of agreement between simulations and experiments was done mostly using stress-strain curves. While this is a good first indicator of agreement, we can not exclude that differently generated initial dislocation microstructures show similar "global" stress-strain behavior with very different "local" "local" dislocation microstructure configurations. If we want to study underlying plastic deformation mechanisms in these situations, it is important that the dislocation microstructure is also comparable to the ones observed in experiments. To this end, we propose using the D2C method which allows us to study and compare these critical local dislocation microstructure features.

In the following, we want to illustrate the application of the D2C method in this manner by studying the effects of prestraining and/or considering cross-slip on the dislocation microstructure in more detail. We simulate the evolution of different initial dislocation microstructures using DDD. Using continuous field descriptions obtained via the D2C method, we quantify the similarity of the average dislocation microstructure between each initial condition. Furthermore, we investigate how similar microstructures within an initial condition category are to each other over the course of the simulations.

5.1. Data generation

In this section, we first cover how the dislocation microstructures were obtained by DDD simulations. These were conducted by Markus Stricker using the DDD code of Daniel Weygand [45, 48]. Subsequently, we outline how we used the D2C method to extract continuous field descriptions of the dislocation microstructures.

Table 5.1.: Parameters used for the discrete dislocation dynamics simulations.

Parameter	Value
lattice constant	0.4045 nm
shear modulus	27 GPa
Poisson's ratio	0.347
tensile axis	[0 1 0]
strain rate	5000 s^{-1}

5.1.1. Generation and evolution of dislocation microstructures

In total, we conducted 40 DDD simulations to investigate the impact of different initial dislocation microstructures and allowed dislocation reactions on the evolution of the microstructures. First, we cover the commonalities of these simulations. The material under consideration is aluminum, which has FCC crystal structure. We used the same parameters as Motz et al. [71], see Table 5.1. The crystallographic axes align with the simulation box axes, i.e., [1 0 0] aligns with the x-axis, [0 1 0] with the y-axis, and [0 0 1] with the z-axis. Each simulation box has a volume of 5 µm × 5 µm × 5 µm. Dislocation loops with radii $2\,\mu m \leq r \leq 8\,\mu m$ were placed randomly with their center in a volume that is four times the size of the simulation box. This way, either whole loops or segments that connect to the surfaces are contained in the simulation box. The dislocations were evenly distributed among all possible slip systems.

At this point, we split the realizations evenly into two sets; one for which we include all dislocation reactions implemented in the DDD code, and another one for which we deactivate cross-slip. The initial dislocation loop configurations were relaxed [71, 73] in all realizations of both sets. During the relaxation, all surfaces were assumed to be traction free. Thus, the driving forces for the dislocations were only their interactions with the free surfaces and amongst themselves. The relaxed dislocation structures exhibited a lower total dislocation density than the initial loop structure. This process of generating a randomized initial loop structure and subsequent relaxation with and without cross-slip was carried out until 20 relaxed dislocation structures with a total dislocation density of roughly $1.15 \cdot 10^{13}\,m^{-2}$ for each set were obtained.

We then split these two sets evenly and prestrained half of the relaxed dislocation microstructures with and without cross-slip. For this, we prescribed fixed boundary conditions on the surface in negative y-direction, and displace-

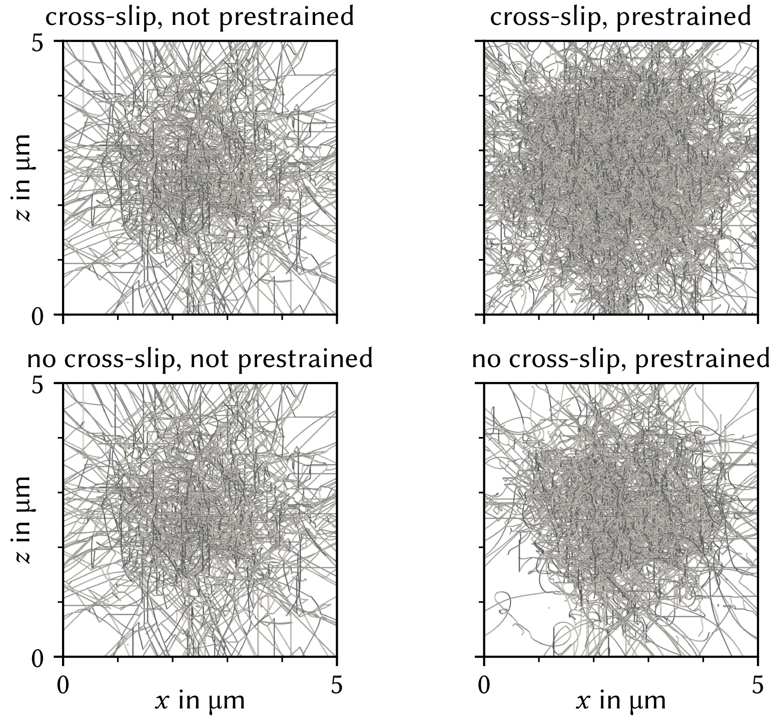


Figure 5.1.: Examples for the initial dislocation microstructures obtained for different combinations of considering cross-slip and prestraining before the tensile test.

ment boundary conditions on the opposite surface in positive y-direction. In this manner, we strained the samples up to a total strain of 0.6 %. Subsequently, we allowed the dislocation microstructure of each prestrained sample to relax.

To summarize, we have generated ten realizations of initial dislocation microstructures in each of four sets;

- relaxed dislocation microstructure with cross-slip,
- relaxed dislocation microstructure without cross-slip,
- prestrained and relaxed dislocation microstructure with cross-slip, and
- prestrained and relaxed dislocation microstructure without cross-slip.

Examples for the initial dislocation microstructure of each set are shown in Figure 5.1. Note that the random distribution of dislocations in the very beginning was the same for all of them. During the initial relaxation, the effect of cross-slip is hardly visible. After prestraining, the dislocation microstructures differ significantly. The one with cross-slip shows a tighter dislocation network that reaches closer to the surfaces than the one without cross-slip.

We then used these different initial dislocation microstructures to perform tensile tests along the y-axis by prescribing displacements on the surface in positive y-direction and fixing the surface in negative y-direction. In this manner, we applied a strain rate of $5000\,\mathrm{s}^{-1}$. Snapshots of the simulations were stored every 50 time steps in the beginning to track the evolution of the dislocation microstructure during the initial deformation in fine detail. Due to the size of the data generated by the DDD framework, we stopped writing output after a set amount of simulation restarts. We continued the simulations until the stresses that had to be applied to impose the prescribed strain rate reached a steady value. The dislocation microstructure at this steady state was also stored.

5.1.2. Extraction of dislocation densities

We extracted the total dislocation density $\rho^{(0)}$ for every simulation snapshot. To this end, we discretized the domain using a mesh consisting of $16 \times 16 \times 16$ elements of equal size. The resulting edge length of each cubic subvolume is $0.3125\,\mu\mathrm{m}$, which is slightly more than the average initial dislocation spacing $\langle \rho^{(0)}_{t=0} \rangle^{-1/2} = 0.2949\,\mu\mathrm{m}$. After extracting the dislocation lines from the results, we applied Jarek's subdivision once before fitting B-spline to them. We did that to account for the DDD code representing dislocations via nodes, see section 3.3.2. For the numerical integration of equation (3.23), we discretized each curve with an average evaluation point spacing of $31.25\,\mathrm{nm}$, i.e., a tenth of the edge length of a subvolume.

5.2. Results

We show the average total dislocation density per initial dislocation microstructure set over time in Figure 5.2. For the realizations that were not prestrained (solid line), we observe that the rate at which the average total dislocation density increases starts to differ strongly after about $0.2\,\mu\mathrm{s}$. After this point in time, the average total dislocation density of realizations that include cross-slip (blue lines) increases faster than that of realization that do not include cross-slip (orange lines). For the realizations with cross-slip, the average total dislocation density reached twice its initial value of $1.15 \cdot 10^{13}\,\mathrm{m}^{-2}$ after about $1\,\mu\mathrm{s}$, whereas it took about $2.4\,\mu\mathrm{s}$ for the ones without cross-slip.

Realizations that were prestrained (dashed lines) exhibit higher and different initial total dislocation densities, $2.91 \cdot 10^{13}\,\mathrm{m}^{-2}$ for realizations with and $1.56 \cdot 10^{13}\,\mathrm{m}^{-2}$ without cross-slip. Upon applying the tensile boundary

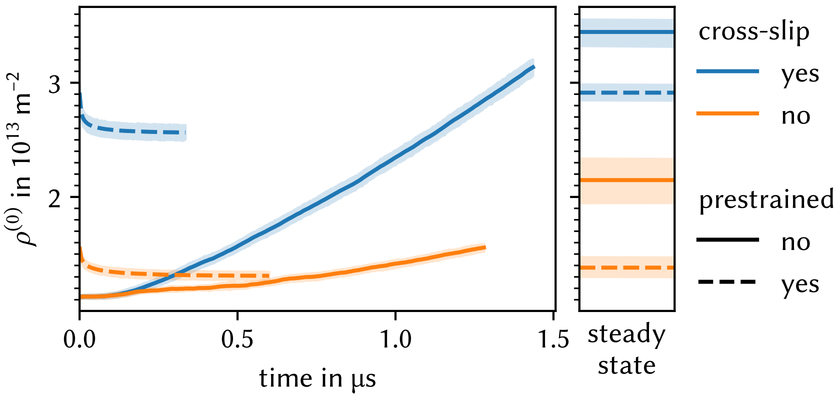


Figure 5.2.: Average total dislocation density $\rho^{(0)}$ of the whole domain over time in the beginning of the simulation (left) and for the steady state (right) for each initial dislocation microstructure set. Steady state refers to the configuration after which no changes in the stress-strain curve are observed. Lines denote the sample mean, and bands represent its 95 % confidence interval.

conditions, the average total dislocation densities decreases in both cases for about 0.1 µs and subsequently plateaus.

Examples for the steady state dislocation microstructures we obtained for each combination of considering cross-slip and applying a prestrain before the tensile test can be seen in Figure 5.3. They are viewed along the tensile direction. The dislocation microstructures we obtained while considering cross-slip are denser and extend closer to the open surfaces when compared to the ones without cross-slip. Between the dislocation microstructures that did not include cross-slip, the one that was prestrained before the tensile test exhibits a stronger dislocation starvation towards the open surface. This difference between applying and not applying a prestrain is almost not visible for the dislocation microstructures that include cross-slip.

We show the probability density functions of the total dislocation density and the curvature for averaged steady state dislocation microstructures in Figure 5.4. They were determined using kernel density estimation. We observe that realizations without cross-slip exhibit a higher probability density for lower total dislocation density and curvature than realizations with cross-slip. When we normalize the total dislocation density by its average we can see that the probability density functions collapse for the same cross-slip state. The differences resulting from different cross-slips activations are that the total dislocation density is slightly more evenly distributed, i.e., there is a lower chance of low density and a higher chance of density values between 1 $\langle\rho^{(0)}\rangle$ to 2.5 $\langle\rho^{(0)}\rangle$.

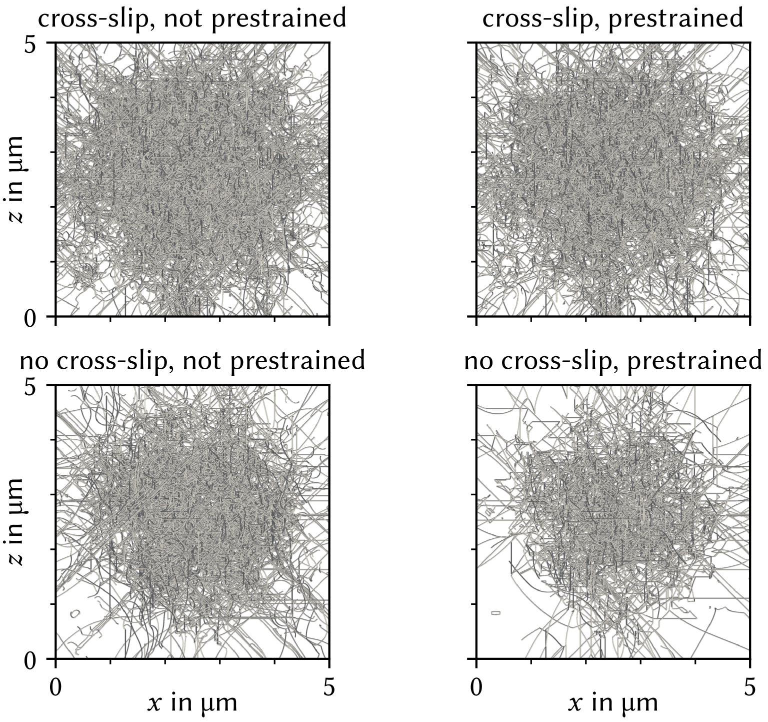


Figure 5.3.: Examples for the steady state dislocation microstructures obtained for different combinations of considering cross-slip and applying a prestrain before the tensile test.

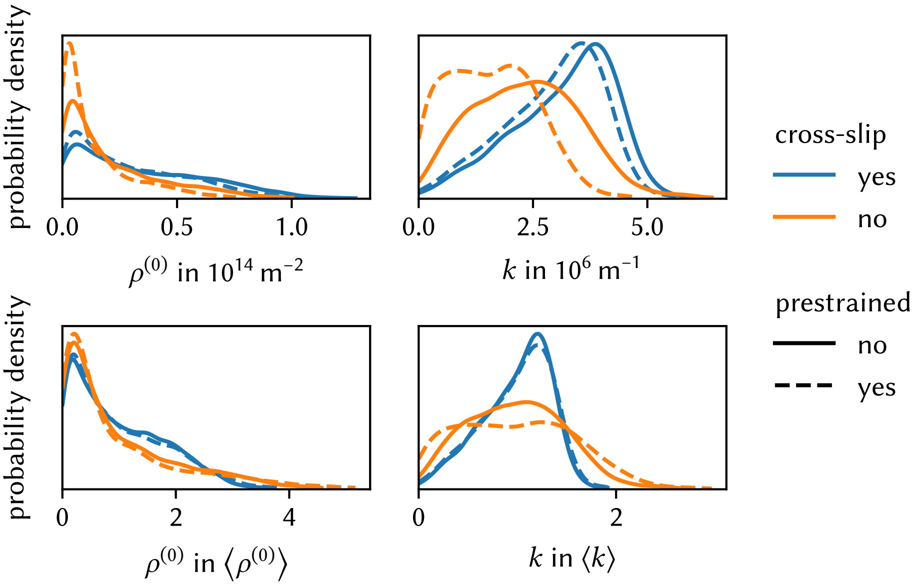


Figure 5.4.: Probability density functions of the total dislocation density and the curvature observed for the averaged steady state configurations of all evolved initial dislocation microstructures. The top row is based on the absolute values, the bottom row is normalized with the sample mean of each set.

In comparison, the distributions of the curvature differ more, especially in their shape. Normalizing the curvature with their average values again collapses the probability density functions for the same state of cross-slip activation. But the shape is more similar for the normalized version of cross-slip enabled realizations. The distribution of the curvature for steady state dislocation microstructures without cross-slip show a more even spread. When cross-slip is considered, we observe a high probability density for curvature values of about $1.2\,\langle k\rangle$.

5.3. Discussion

5.3.1. Influence of cross-slip on the total dislocation density

The main differences in the average total dislocation density configurations with respect to enabling cross-slip are an increase in total dislocation density and higher total dislocation density closer to the open surfaces. The latter observation can be seen in Figure 5.5. It shows the total dislocation density

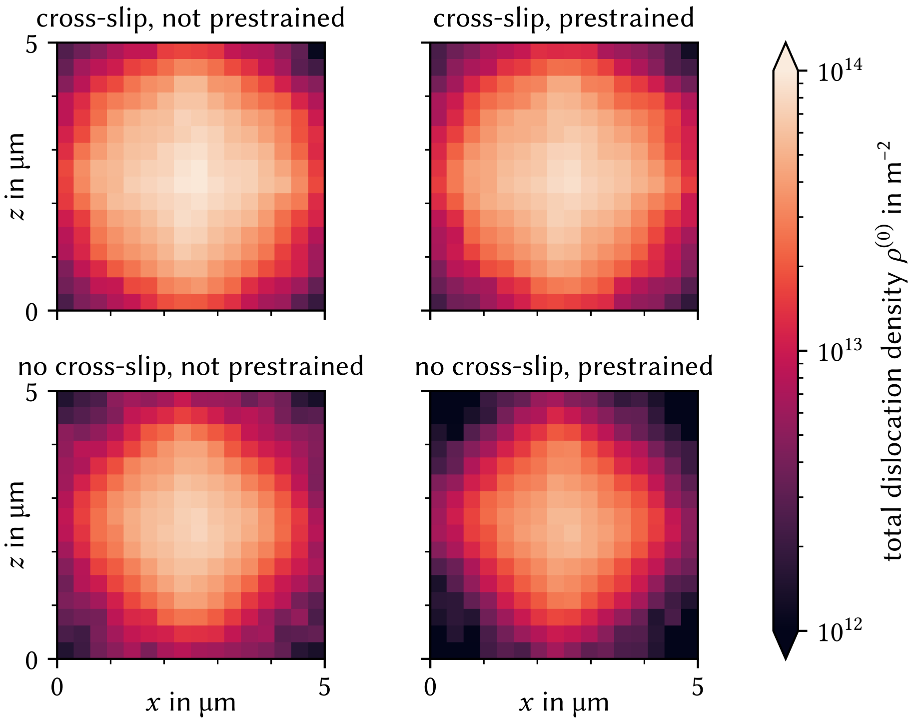


Figure 5.5.: Total dislocation density $\rho^{(0)}$ averaged along the y-axis, i.e., the tensile axis, for each type of initial dislocation microstructure.

of the “mean dislocation microstructure” of each set of initial dislocation microstructures, where the mean is computed via equation (3.33). This mean is then averaged along the *y*-axis to yield the information shown in the figure. Regardless of the initial dislocation microstructure, we can see a maximum of the total dislocation density in the center. We observe a decrease in the average total dislocation density with increasing distance from the center. This gradient is strongest for realizations without cross-slip that were prestrained. Without the prestraining, this gradient is less pronounced. Realizations that include cross-slip on the other hand show higher average total dislocation densities towards the open surfaces. cross-slip is contributing to this in two ways.

The first reason is that cross-slip adds degrees of freedom to the ways a dislocation can propagate within the material. As dislocations propagate, their line length increasing and so does the total dislocation density. These dislocations interact with each other via their stress fields and potentially forming junctions. Some of these dislocations might be hindered from propagation due these effects. If cross-slip is possible, parts of dislocations with screw character can switch their slip plane to one that does not exhibit the same barriers and potentially propagate further. This change of slip plane also means that a dislocation is not confined to a single plane in space. Via successive cross-slip, the motion of dislocations becomes three-dimensional. Therefore a more even distribution of dislocation in the material is possible, leading to higher average total dislocation densities that is also more homogeneous.

But if dislocations are able to move more freely, should they not also be able to better reach the open surfaces that they are attracted by? This is where the second contribution of cross-slip comes into play. Yes, cross-slip increases the possible ways of propagating. But, at the same time the dislocation motion at the sites where cross-slip took place is constrained to the intersection line of the two slip planes across which cross-slip occurred. Thus, the potential of moving three-dimensionally comes with the drawback of having parts that are only able to move in one dimension. These constrained parts of the dislocation are then acting as possible obstacles for other dislocations, or get pinned easier by other barriers which a regular dislocation line might be able to overcome otherwise. And this hinders dislocations from leaving the material through the open surfaces.

To summarize, cross-slip increases the volume available to dislocations to occupy while at the same time introducing more obstacles for parts of the dislocations to get stuck at. This combination leads to a higher, more evenly spread total dislocation density, even close to open surfaces.

5.3.2. History dependence of the dislocation microstructures

By comparing realizations that were prestrained to realizations that were not prestrained, we essentially probe whether the final dislocation microstructure depends on the history of the material. If the dependence was small, their dislocation microstructure should be comparable as more strain is applied.

If cross-slip is considered, we observe that the probability density functions of both the total dislocation density as well as the curvature of the dislocations are close to each other. When normalized by their respective means, they also collapse nicely on top of each other. To a certain extent, we also observe this for the probability density functions of the dislocation microstructures without cross-slip. While the shape is somewhat similar, especially for the total dislocation density, we see larger differences between the mean value of the total dislocation density and the overall shape of the curvature distribution.

This is not the case for realizations without cross-slip. Here, the dislocation microstructure shows a stronger dependence on whether it was obtained from prestrained realizations or not. The first indicator are the probability density functions of the total dislocation density and the curvature. While the shapes of the probability density functions are somewhat similar when we normalize them by their respective averages, it shows that the baseline of total dislocation density between prestrained and not prestrained realizations differs a lot compared to the realizations with cross-slip.

But, this only tells us about their overall distribution and does not take the spatial configuration into account. When we look at the spatial configuration, we see the resulting lower average total dislocation density of prestrained realizations, as well as fewer dislocations that remain close to the free surfaces.

As we can see in Figure 5.5, these differences in the distributions are not arbitrarily distributed in space. We clearly observe that the differences between realizations for which cross-slip was not considered are most pronounced in the vicinity of the open surfaces. In a database setting with potentially more sets of dislocation microstructures to compare, we desire to have a single, scalar measure of the similarity that takes the spatial configuration into account. To this end, we compute the CV of the MAE and CV of the RMSE for each pair of set-wise averaged final dislocation microstructures via equations (3.31) and (3.32), respectively. They are shown in Figure 5.6. As outlined in section 3.3.3, this allows us to quantify how different two dislocation microstructures are via a scalar, dimensionless value. We observe that the most similar “average dislocation microstructures” are the ones for which cross-slip was taken into account. The next closest pair are the prestrained dislocation

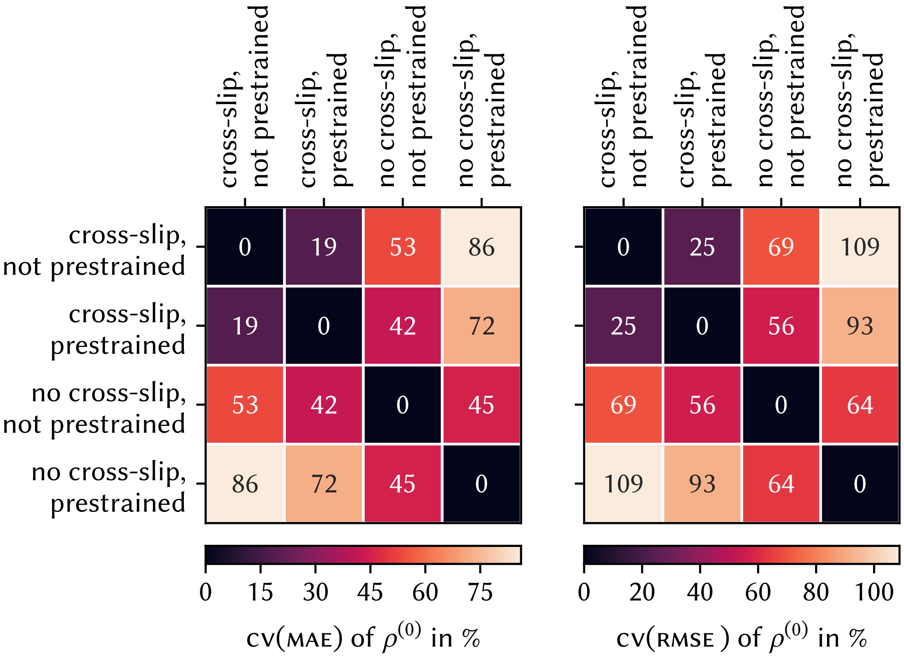


Figure 5.6.: Pairwise coefficient of variation of the mean absolute error (left) and coefficient of variation of the root-mean-square error (right) for the averaged total dislocation density of the steady state dislocation microstructures evolved from differently obtained and evolved initial microstructures. A higher value denotes less similarity.

microstructures with enabled cross-slip, and the dislocation microstructures that were not prestrained and did not consider cross-slip. Thus, taking the spatial configuration into account, we can see that the similarities of the dislocation microstructures is not limited to just the overall distribution alone, but is also true for the spatial configuration.

We can explain these differences in the final dislocation microstructure by the aforementioned degree of freedom which cross-slip adds to the dislocation motion. When we assume the final dislocation microstructure of realizations with cross-slip to represent an energetically favorable one, then cross-slip enables the dislocation microstructure to form this configuration independently of the initial microstructure. On the other hand, the limitations put on the dislocation motion by not enabling cross-slip results in a larger dependence on the initial microstructure and therefore the history of the material. All of these considerations are condensed by the CV of the root-mean-square deviation (RMSD) for the average total dislocation density. These values are able to capture our observations in scalar, dimensionless quantities that hence may be used as measures for the history dependence of dislocation microstructures for which cross-slip was either activated or deactivated within this study.

5.3.3. Similarity of dislocation microstructures over time

We now investigate the temporal evolution of differences within each initial dislocation microstructure set to their average. To this end, we compute the CV of the standard deviation of the sample $\mathrm{CV}(s)$ via equation (3.32) for each set as a function of time. Smaller values indicate that the dislocation microstructures sharing the same type of initial configuration are more similar to each other on average. We show these values in Figure 5.7.

First, we consider the evolution of realizations that were not prestrained. Initially, their realizations show the same variation to their average with a value slightly larger than 1.0. During the first 0.2 µs, the value decreases slightly, i.e., the dislocation microstructures of these two sets become more similar to their average. Subsequently, the dislocation microstructures that include cross-slip continue to become more and more similar to their average, whereas the ones that do not include cross-slip start to differ more from their mean. After increasing until about 0.6 µs into the simulation, the variation of the latter plateaus and starts to decrease slightly after 1.0 µs.

The dislocation microstructures of prestrained samples exhibit different initial variations around their averages. Realizations whose evolution included cross-slip start out at about $\mathrm{CV}(s) = 0.85$, whereas realizations that did not include it start out at approximately $\mathrm{CV}(s) = 1.04$. The variation initially

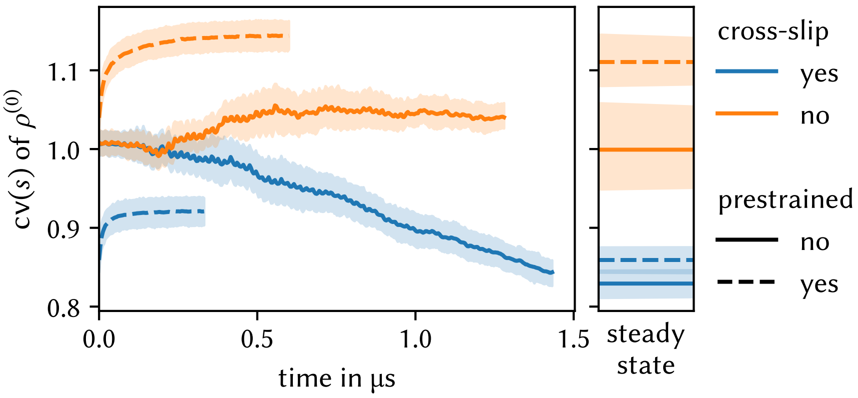


Figure 5.7.: Temporal evolution of the coefficient of variation of the standard deviation of the sample of the total dislocation density for each set of initial dislocation microstructure. Steady state refers to the configuration after which no changes in the stress-strain curve are observed. Lines denote the mean value, bands the 95 % confidence interval.

increases in both cases. For microstructures including cross-slip a plateau is reached shortly after the beginning of the simulation at 0.92. The ones not including cross-slip continue to grow more dissimilar, but at a slower rate than during their initial increase in variation.

Overall, the consideration of cross-slip changes the implications for whether prestraining a sample affects the evolution of a dislocation microstructure. If cross-slip is not considered, special care has to be taken during the generation of the initial dislocation microstructure as the difference can be measured over a long span of simulation time The initial structure affects the pathways of dislocations strongly and therefore affects the possible ways of the microstructure to evolve. Less care is required when cross-slip is taken into account and sufficient simulation time is allowed to pass. These observations underline the aforementioned effects of cross-slip. The additional degree of freedom for dislocations enables the dislocation microstructure to form more similar, possibly more stable, configurations over time.

5.4. Summary and conclusion

We have generated four different sets of initial dislocation microstructures and have simulated their evolution in time during a tensile test while either considering or neglecting cross-slip. We then extracted dislocation microstructure features. Based on these features, we compared the different sets of dislocation microstructures with and without taking the spatial configuration of the features into account. We were able to gain insight into the effect that cross-slip has on the history dependence of the dislocation microstructure configuration. Vie the additional degree of freedom it provides for the motion of dislocations, it reduces the effect an intermediate relaxation has upon the final microstructure. Furthermore, we studied the impact of cross-slip on the statistical similarity of dislocation microstructures depending on how they were generated.

Given that we see these effects and are able to understand them, we conclude that the D2C method is well suited to make dislocation microstructures comparable when the spatial configuration of the microstructure is important. In combination with statistical measures like the CV, we see potential for using it on larger studies in the context of a dislocation microstructure database.

Furthermore, we conclude that cross-slip is an important mechanism to consider the evolution of dislocation microstructures. It acts as an additional degree of freedom that allows the dislocations to form possibly more stable configurations less dependent on the initial configuration. And while cross-slip might increase the computational cost of DDD simulations, the fact that the dislocation microstructures grow more similar over time indicates that fewer simulations have to be performed in order to capture statistical aspects of the systems.

6. Machine-learning-based classification of dislocation microstructures

In the previous chapter, we have utilized continuum dislocation fields to determine average dislocation microstructures, and to compare dislocation microstructures with each other. This was all performed manually and guided by the authors of this work. Eventually, we want to assemble more data on dislocation microstructures and use statistical approaches to identify patterns across experiments and simulations, possibly to predict properties of a material based on its dislocation microstructure. The inherent complexity of dislocation microstructures makes this a difficult undertaking. But in recent years, the resurgence of machine learning in combination with better means of characterizing and analyzing microstructures has lead to improvements in predicting, e.g., the mechanical properties of microstructures[1] [74, 75]. If similar techniques were employed for dislocation microstructures, our understanding of their collective contribution to plastic deformation could be expanded.

An important task that precedes the actual application of machine learning models, and contributes significantly to their success, is the extraction of suitable input variables. In this chapter, we want to study whether the continuum dislocation fields extracted via the D2C method are suitable features. To this end, we conduct two numerical experiments during which dislocation microstructures are evolved using DDD simulations.

First, we allow dislocation microstructures that were initialized randomly within cube-shaped nanoparticles of different sizes to relax. Based only on features extracted from the dislocation microstructure close to the surfaces of the nanoparticles, we try to determine the size of the particle. We expect the final dislocation microstructures to contain mostly SSDs in this case.

Second, we evolve dislocation microstructures during the indentation of nanoparticles of different sizes. Similarly to the first experiment, we want to

---

[1] Note that these microstructure in this case does not refer to dislocation microstructures specifically, but to the grain or phase configuration.

identify the particle size relying only on features obtained from the dislocation microstructure under the indenter. In this case, we expect the dislocation microstructures to comprise mostly GNDs.

These two experiments cover two extreme cases of dislocation microstructures and can be seen as benchmarks for the extraction of dislocation microstructure features. We first cover the commonalities for both studies, i.e., the employed continuum dislocation field combinations, and the machine learning algorithms we base the machine learning models on. For each experiment, we then outline how we generated and evolved the dislocation microstructures, as well as how we used the D2C method to extract features from the microstructures. Subsequently, we summarize the results and discuss the ones specific to its experiment. We then discuss the common observations of the two experiments. And we summarize our findings and conclude this chapter.

6.1. Commonalities between the numerical experiments

6.1.1. Employed continuum dislocation field combinations

To understand how different continuum dislocation fields affect the performance of the machine learning models, several different field combinations are used to train and test them. The combinations of fields used for building and evaluating a machine learning model are denoted by their symbols enclosed in curly braces, i.e., $\{\rho^{(0)}, q^{(0)}\}$ means that both, the 0th-order dislocation density alignment tensor and the dislocation curvature density are used as features.

6.1.1.1. Single fields

Single fields of the CDD framework (see 2.2.5.4) up to the 5th-order, i.e., $\{\rho^{(0)}\}$, $\{\boldsymbol{\rho}^{(1)}\}$, $\{\boldsymbol{\rho}^{(2)}\}$, $\{\boldsymbol{\rho}^{(3)}\}$, $\{\boldsymbol{\rho}^{(4)}\}$, and $\{\boldsymbol{\rho}^{(5)}\}$, as well as the dislocation curvature density $\{q^{(0)}\}$, and the Kröner–Nye tensor $\{\boldsymbol{\alpha}\}$ (see 2.2.5.2) are used to probe the effect of a single field onto the quality of the machine learning models.

6.1.1.2. Continuum dislocation dynamics field pairs

Due to (2.33), we known that two CDD fields of successive order contain the entire information of all lower order fields. Thus, we consider combinations of two successive fields up to the 5th-order, i.e., $\{\rho^{(0)}, \boldsymbol{\rho}^{(1)}\}$, $\{\boldsymbol{\rho}^{(1)}, \boldsymbol{\rho}^{(2)}\}$, $\{\boldsymbol{\rho}^{(2)}, \boldsymbol{\rho}^{(3)}\}$, $\{\boldsymbol{\rho}^{(3)}, \boldsymbol{\rho}^{(4)}\}$, and $\{\boldsymbol{\rho}^{(4)}, \boldsymbol{\rho}^{(5)}\}$, as well to see how including higher orders of the dislocation density alignment tensor affects the machine learning models.

6.1.1.3. Continuum dislocation dynamics field pairs with dislocation curvature density

Within the CDD framework, the dislocation curvature density $q^{(0)}$ is required in addition to aforementioned successive-order fields to accurately evolve of dislocation densities. For this reason, we take combinations of two successive fields up to the 5th-order with the dislocation curvature density, i.e., $\{\rho^{(0)}, \boldsymbol{\rho}^{(1)}, q^{(0)}\}$, $\{\boldsymbol{\rho}^{(1)}, \boldsymbol{\rho}^{(2)}, q^{(0)}\}$, $\{\boldsymbol{\rho}^{(2)}, \boldsymbol{\rho}^{(3)}, q^{(0)}\}$, $\{\boldsymbol{\rho}^{(3)}, \boldsymbol{\rho}^{(4)}, q^{(0)}\}$, and $\{\boldsymbol{\rho}^{(4)}, \boldsymbol{\rho}^{(5)}, q^{(0)}\}$, into account to examine the impact of the dislocation curvature density onto the machine learning models.

6.1.1.4. Continuum dislocation dynamics full fields

As the machine learning models are not explicitly informed about the relationship (2.33), providing the lower-order fields in addition to the two successive highest-order fields may affect the machine learning model quality. To test this, we consider additional combinations including all CDD fields up the 5th-th order, i.e., $\{\rho^{(0)}, \boldsymbol{\rho}^{(1)}, \boldsymbol{\rho}^{(2)}\}$, $\{\rho^{(0)}, \boldsymbol{\rho}^{(1)}, \boldsymbol{\rho}^{(2)}, \boldsymbol{\rho}^{(3)}\}$, $\{\rho^{(0)}, \boldsymbol{\rho}^{(1)}, \boldsymbol{\rho}^{(2)}, \boldsymbol{\rho}^{(3)}, \boldsymbol{\rho}^{(4)}\}$, and $\{\rho^{(0)}, \boldsymbol{\rho}^{(1)}, \boldsymbol{\rho}^{(2)}, \boldsymbol{\rho}^{(3)}, \boldsymbol{\rho}^{(4)}, \boldsymbol{\rho}^{(5)}\}$.

6.1.1.5. Continuum dislocation dynamics full fields with dislocation curvature density

As before, we also want to take these CDD field combinations in conjunction with the dislocation curvature density $q^{(0)}$ into account, i.e., $\{\rho^{(0)}, \boldsymbol{\rho}^{(1)}, \boldsymbol{\rho}^{(2)}, q^{(0)}\}$, $\{\rho^{(0)}, \boldsymbol{\rho}^{(1)}, \boldsymbol{\rho}^{(2)}, \boldsymbol{\rho}^{(3)}, q^{(0)}\}$, $\{\rho^{(0)}, \boldsymbol{\rho}^{(1)}, \boldsymbol{\rho}^{(2)}, \boldsymbol{\rho}^{(3)}, \boldsymbol{\rho}^{(4)}, q^{(0)}\}$, and $\{\rho^{(0)}, \boldsymbol{\rho}^{(1)}, \boldsymbol{\rho}^{(2)}, \boldsymbol{\rho}^{(3)}, \boldsymbol{\rho}^{(4)}, \boldsymbol{\rho}^{(5)}, q^{(0)}\}$.

6.1.1.6. Continuum dislocation dynamics fields with Kröner–Nye tensor

Because the deformation character of dislocations on different slip systems is not taken into account, field combinations of previous types containing at least 1st-order up to 2nd-order are added while replacing the “density of GNDs” with the Kröner–Nye tensor $\boldsymbol{\alpha}$, i.e., $\{\rho^{(0)}, \boldsymbol{\alpha}\}$, $\{\rho^{(0)}, \boldsymbol{\alpha}, q^{(0)}\}$, $\{\rho^{(0)}, \boldsymbol{\alpha}, \boldsymbol{\rho}^{(2)}\}$, and

$\{\rho^{(0)}, \boldsymbol{\alpha}, \boldsymbol{\rho}^{(2)}, q^{(0)}\}$. With this, we may explore the effect of partially incorporating the full character of dislocations on the quality of the machine learning models.

6.1.1.7. Continuum dislocation dynamics orientation distribution fields

For field combinations including CDD fields up to the 2nd-order, combinations where these are replaced by their corresponding orientation distribution $\boldsymbol{\rho}^{(n)}/\rho^{(0)}$ are taken into account to evaluate the usage of the local orientation distribution instead of the absolute density values.

6.1.1.8. Dislocation feature density tensor fields

To study the effect of explicitly including the Burgers vector information, we also consider the single field combinations $\{\boldsymbol{\rho}^{(1,0)}\}$, $\{\boldsymbol{\rho}^{(1,2)}\}$, $\{\boldsymbol{\rho}^{(1,3)}\}$, $\{\boldsymbol{\rho}^{(2,0)}\}$, $\{\boldsymbol{\rho}^{(2,1)}\}$, and $\{\boldsymbol{\rho}^{(2,2)}\}$. Note that we already consider $\boldsymbol{\rho}^{(1,1)}$, as it is equivalent to the Kröner–Nye tensor $\boldsymbol{\alpha}$. Similarly to the CDD fields, we also include pairs $\{\boldsymbol{\rho}^{(1,0)}, \boldsymbol{\alpha}\}$, $\{\boldsymbol{\alpha}, \boldsymbol{\rho}^{(1,2)}\}$, and $\{\boldsymbol{\rho}^{(1,2)}, \boldsymbol{\rho}^{(1,3)}\}$.

6.1.2. Employed machine learning algorithms

All aforementioned field combinations are used in combination with each machine learning algorithm outlined in section 3.4, i.e., GNB, SGD, and HGBM. With this we study how different machine learning algorithms make use of the features extracted via the D2C method.

Note that we do not tune the hyperparameters of the resulting machine learning models, but instead use the default values provided by scikit-learn These are carefully provided by the developers to give reasonably good results for a variety of applications. While their optimization leads to better machine learning model performance, it would require a fair amount of additional computational resources, and would therefore limit the number of continuum dislocation field combinations we could study. As this work primarily focuses on the effect of different field combinations, we have chosen not to optimize the hyperparameters in favor of considering more field combinations.

All features are normalized such that the sample mean and sample standard deviation on the training data set are zero and one, respectively. Even though this should not be required for the GNB and HGBM machine learning algorithms, the actual implementation of the GNB within scikit-learn requires us to do so. For numerical stability, a portion of the largest sample variance is added to the sample variance of all other features as well. Because the features obtained from different fields exhibit values in vastly different orders of magnitude, this can lead to features being more or less ignored. To make sure that this does not affect the machine learning models studied in this work, we therefore decided to normalize the input for all of them.

6.2. Relaxation

The interactions of dislocations with surfaces become increasingly important for smaller specimen like the ones used for TEM. This is due to every dislocation being close to an open surface that they are attracted by. As the attractive force exerted on a dislocation close to a free surface depends on their distance to each other, we may expect dislocation microstructures of nanoparticles to differ for various particle sizes. We expect to see lower dislocation densities towards the surfaces, with the gradient depending on the size of the particle. In the following, we conduct a numerical experiment and try to train machine learning models to classify the size of the particle a dislocation microstructure belongs to, solely based on features of this microstructure in the vicinity of an open surface.

6.2.1. Data generation

The data was generated in three steps. First, we generated and evolved the dislocation microstructure using the MODEL DDD code [49, 46]. After the microstructure was completely relaxed, we extracted features via the D2C method. We then used these features as input to train machine learning models and evaluate their results. These steps are outlined in the following.

6.2.1.1. Generation and evolution of dislocation microstructures

We filled cube-shaped specimens with edge lengths of 30 nm, 60 nm and 90 nm with randomly placed dipolar edge loops on all slip systems until we reached a total dislocation density of approximately $5 \cdot 10^{16}\,\mathrm{m}^{-2}$. The FCC crystal structure was aligned with the coordinate axes, i.e., the [1 0 0] direction coincides with the x-axis, [0 1 0] with the y-axis, and [0 0 1] with the z-axis.

Subsequently, we allowed the microstructure to relax without application of an external load. For simplicity and computational performance reasons, we considered neither the formation of junctions nor cross-slip. We took image forces due to the open surfaces into account. Therefore, the main driving

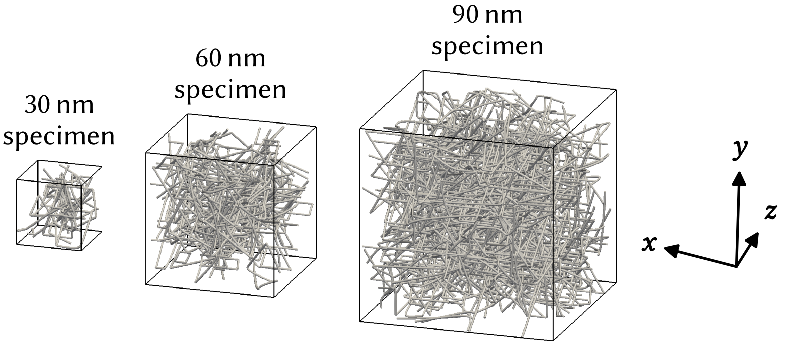


Figure 6.1.: Examples for dislocation microstructures obtained from relaxation simulations for different specimen sizes.

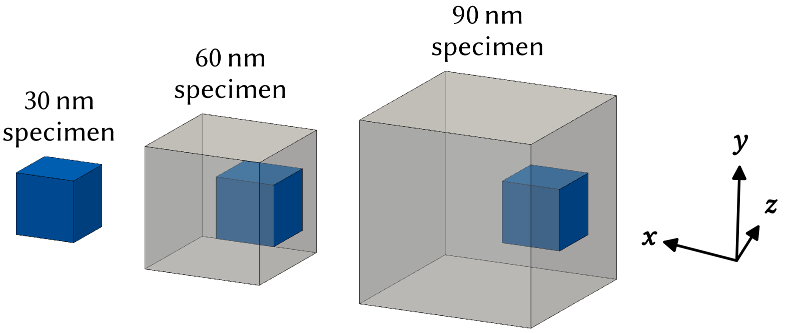


Figure 6.2.: Specimens (gray) and one of the locations of a sample (blue) whose dislocation microstructure are considered for the machine learning model. Regardless of the specimen size, the size of the sample is always 30 nm.

forces for dislocation motion were dislocation-dislocation and dislocation-surface interactions. In total, we simulated 8202 specimens for each particle size. Exemplary dislocation microstructures are shown in Figure 6.1.

6.2.1.2. Extraction of dislocation microstructure features

As we constrained ourselves to the dislocation microstructure close to a surface, we only considered cube-shaped samples with an edge length of 30 nm at the surface centers. The specimens and the location of one of the samples within each specimen can be seen in Figure 6.2. Due to symmetries of the crystal structure, the specimen and the sample, we were able to augment the number of observations. One sample was taken from each of the six side of the specimens and rotated so that the open surface is oriented towards the positive direction of the x-axis. The four-fold rotational and mirror symmetries of the FCC crystal structure then enabled us to generate eight observations from each of the six samples taken from one specimen, resulting in a total of 48 observations per specimen. Overall, we obtained 393 696 observations per particle size.

We discretized the samples into a regular grid with the same number of voxels along each direction, ranging from 1 to 6. As the MODEL DDD code uses cubic Hermite splines to represent dislocations, the direct computation of quantities required within the D2C method was possible. We performed the numerical integration of equation (3.23) using a uniform evaluation point density of $0.3\,\mathrm{nm}^{-1}$ along the curves, i.e., at least 16 evaluation points per voxel for the finest spatial discretization. Only independent components of (partially) symmetric tensors were used to avoid duplicate features for the machine learning models.

6.2.1.3. Machine learning

To evaluate the performance of the machine learning models, we performed five shuffled stratified five-fold cross-validations for each combination of spatial discretization and dislocation microstructure features. We also computed learning curves to better understand how many observations are required to sufficiently train the machine learning models. Furthermore, we determined the permutation feature importance for each dislocation density field to gain insight into how the machine learning models make use of the data available to them. In addition to that, we also computed the permutation feature importance for the distance of the voxels from the surface, irrespective of the fields. This was done to see *where* the distinct differences between the dislocation microstructures of different specimen sizes are.

6.2.2. Results

6.2.2.1. Average total dislocation densities

The average total dislocation densities within the sample for all spatial discretizations and specimen sizes are shown in Figure 6.3. With an average total dislocation density of $4.59 \cdot 10^{16}\,\mathrm{m}^{-2}$, the samples of the 30 nm specimen exhibit the lowest value out of all specimen sizes. The next higher value is observed for the samples taken from the 90 nm specimen with $7.91 \cdot 10^{16}\,\mathrm{m}^{-2}$. The samples taken from the 60 nm specimen show the largest average total dislocation density with $8.79 \cdot 10^{16}\,\mathrm{m}^{-2}$.

Finer spatial discretizations reveal the distribution of the average total dislocation density within the samples of each specimen size. As the voxels get

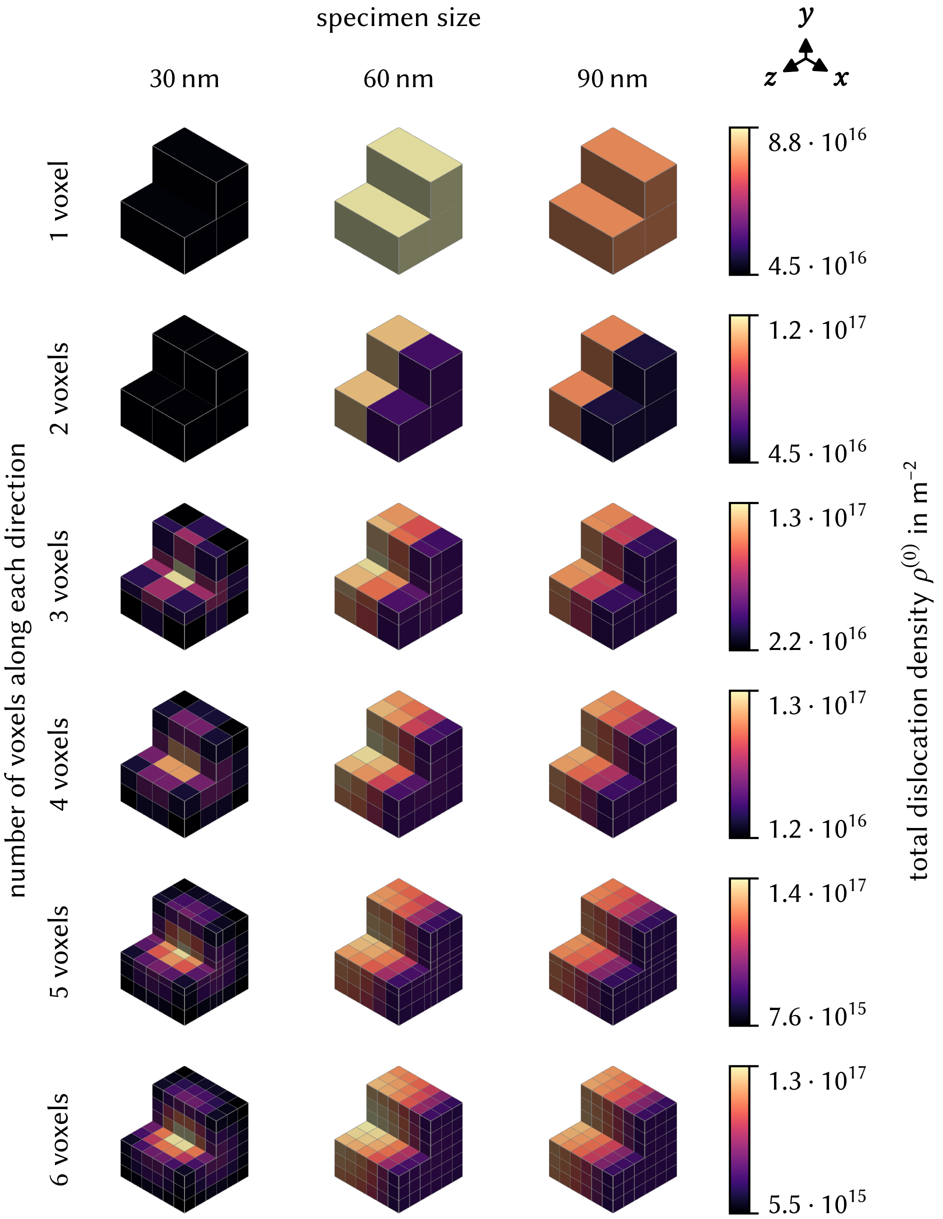


Figure 6.3.: Average total dislocation densities for each specimen size and spatial discretization. A cut was taken from each sample to reveal the values at the center. The open surface common to all specimen sizes is along the positive x-direction. Colorbars account for the entire value range within their respective row.

smaller, the range of observed average total dislocation density values increases. This is due to localized density fluctuations being averaged out for larger voxels. Regardless of the specimen size, the average total dislocation density exhibits their lowest values close to the open surface in positive x-direction. The values increase with further distance from the open surface. For the samples taken from 30 nm specimen, a decrease of the average total dislocation density is observed along all directions from the center of the samples. In samples taken from the larger specimen, the gradient is monotonically increasing with smaller x-coordinate. Compared to the samples of the 90 nm samples, we can also observe a decrease of the average total dislocation density along the y- and x-direction within the 60 nm samples for increasing distance from the symmetry axis.

6.2.2.2. Performance of the machine learning models

The accuracy scores for different field combinations and different spatial discretizations are shown in Figure 6.4 for GNB, in Figure 6.5 for SGD, and in Figure 6.6 for HGBM.

Gaussian naïve Bayes

For the models based on the GNB algorithm, the single field performance ranked from worst to best is $q^{(0)}$, $\boldsymbol{\rho}^{(1)}$, $\boldsymbol{\alpha}$, $\boldsymbol{\rho}^{(5)}$, $\boldsymbol{\rho}^{(3)}$, $\rho^{(0)}$, $\boldsymbol{\rho}^{(2)}$, and $\boldsymbol{\rho}^{(4)}$. Overall, including higher-order CDD fields positively affects the quality of the model. For field combinations that include at least the 4th-order CDD field, only including the two highest-order fields instead of all the lower-order fields as well results in higher accuracy scores. Including lower-order fields is better when the maximum included order is smaller than 4. Using the curvature density results in worse model quality. Replacing the 1st-order CDD field $\boldsymbol{\rho}^{(1)}$ with the Kröner–Nye tensor $\boldsymbol{\alpha}$ improves the model. With the exception of field combinations that include at most two fields of low orders, a coarser spatial discretization results in better performance of the model. Combinations including only dislocation feature density tensors do not perform as well as the ones including only dislocation density alignment tensors.

For at most two voxels along each direction, using $q^{(0)}$ or $\boldsymbol{\rho}^{(1,0)}$ with one voxels as single field results in an accuracy score of at most 0.5, which compared to other accuracy scores within the GNB-based machine learning models is particularly low.

Stochastic gradient descent

For the models based on the SGD algorithm, the single field performance ranked from worst to best is $\boldsymbol{\alpha}$, $\boldsymbol{\rho}^{(1)}$, $\boldsymbol{\rho}^{(3)}$, $\boldsymbol{\rho}^{(5)}$, $q^{(0)}$, $\rho^{(0)}$, $\boldsymbol{\rho}^{(2)}$, and $\boldsymbol{\rho}^{(4)}$. Including higher-

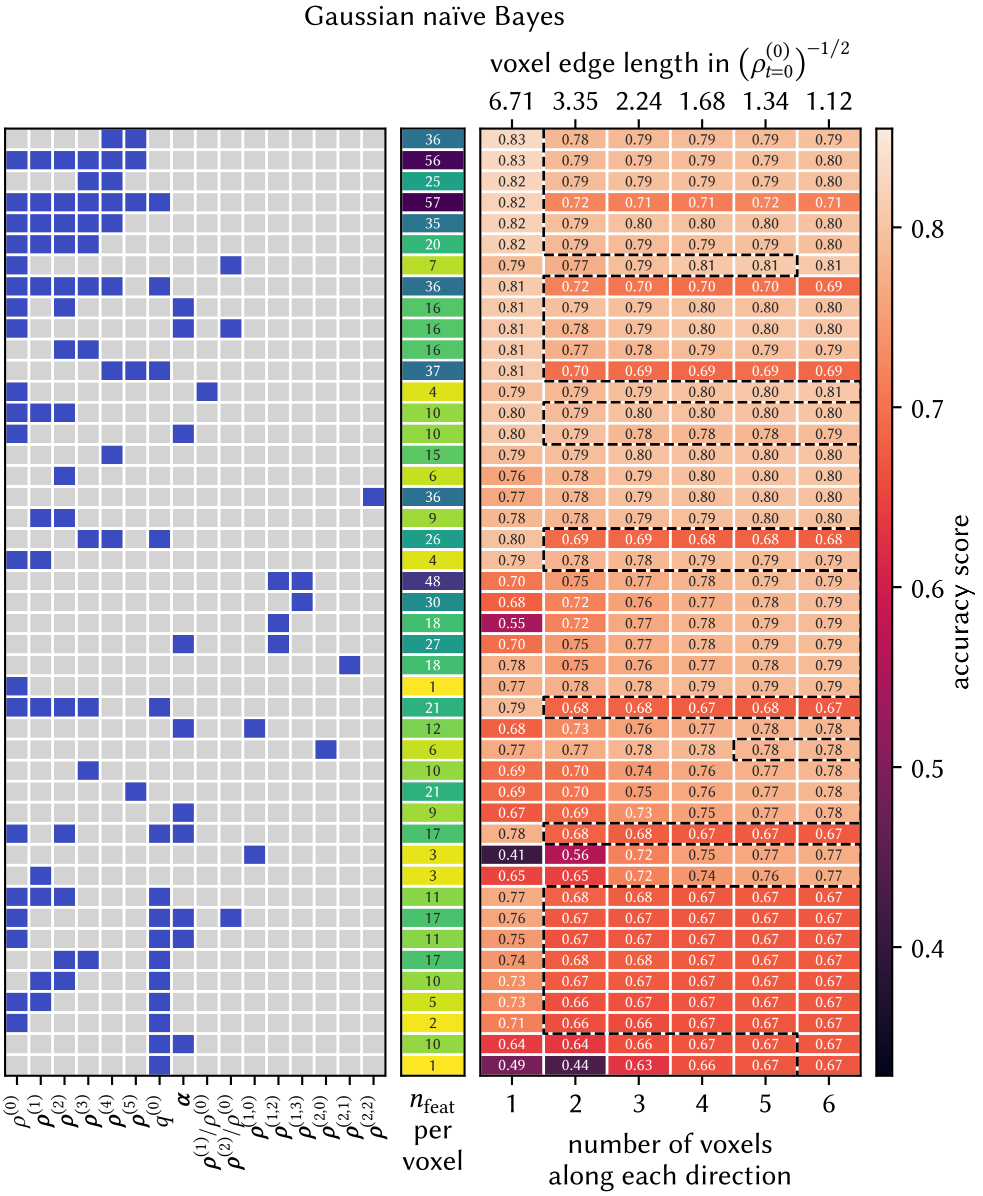


Figure 6.4.: Accuracy scores of the GNB-based machine learning models for different field combinations and different spatial discretizations. Rows are ordered by the highest achieved by the respective field combination. The dashed line indicates the boundary for which an increase in spatial resolution results in an increased .

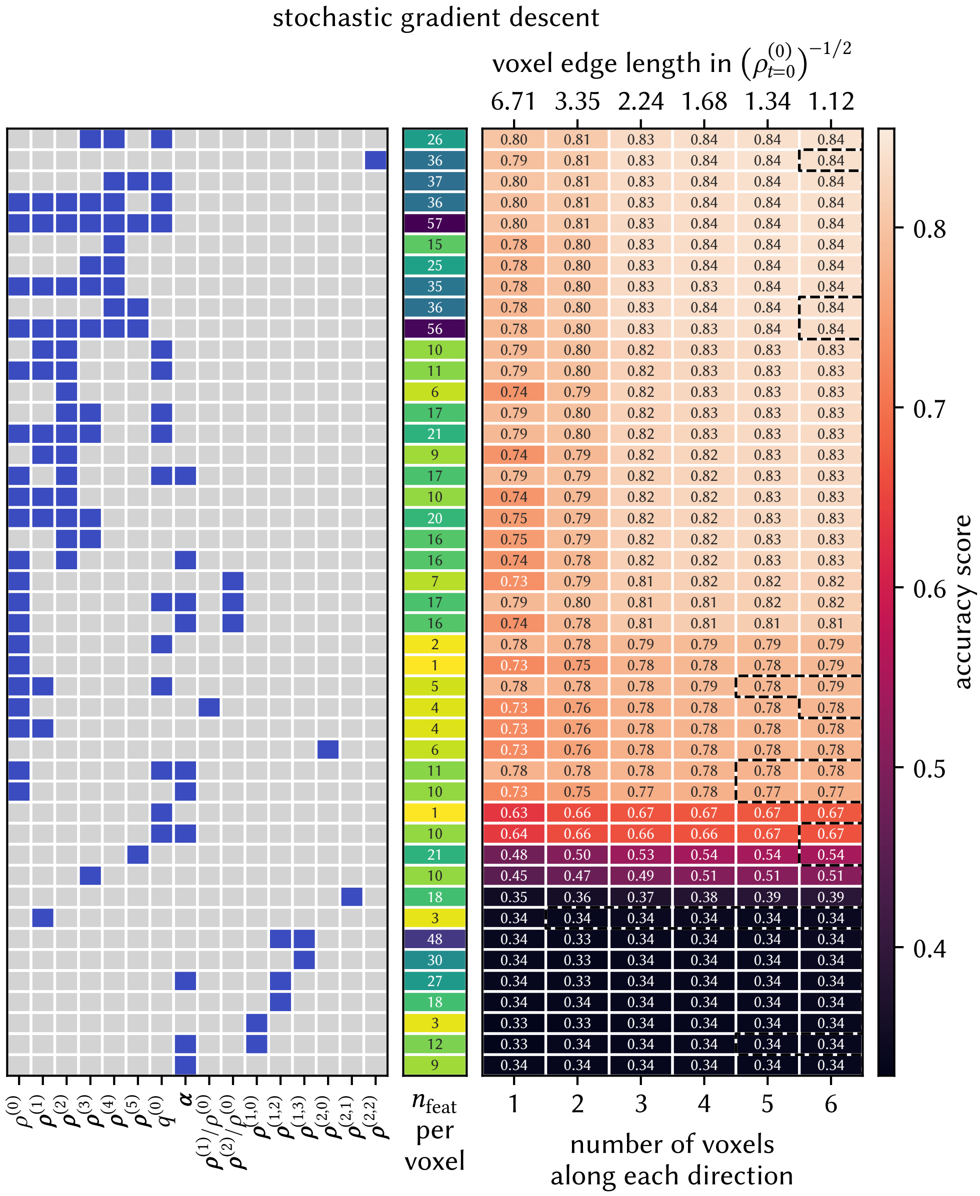


Figure 6.5.: Accuracy scores of the SGD-based machine learning models for different field combinations and different spatial discretizations. Rows are ordered by the highest achieved by the respective field combination. The dashed line indicates the boundary for which an increase in spatial resolution results in an increased .

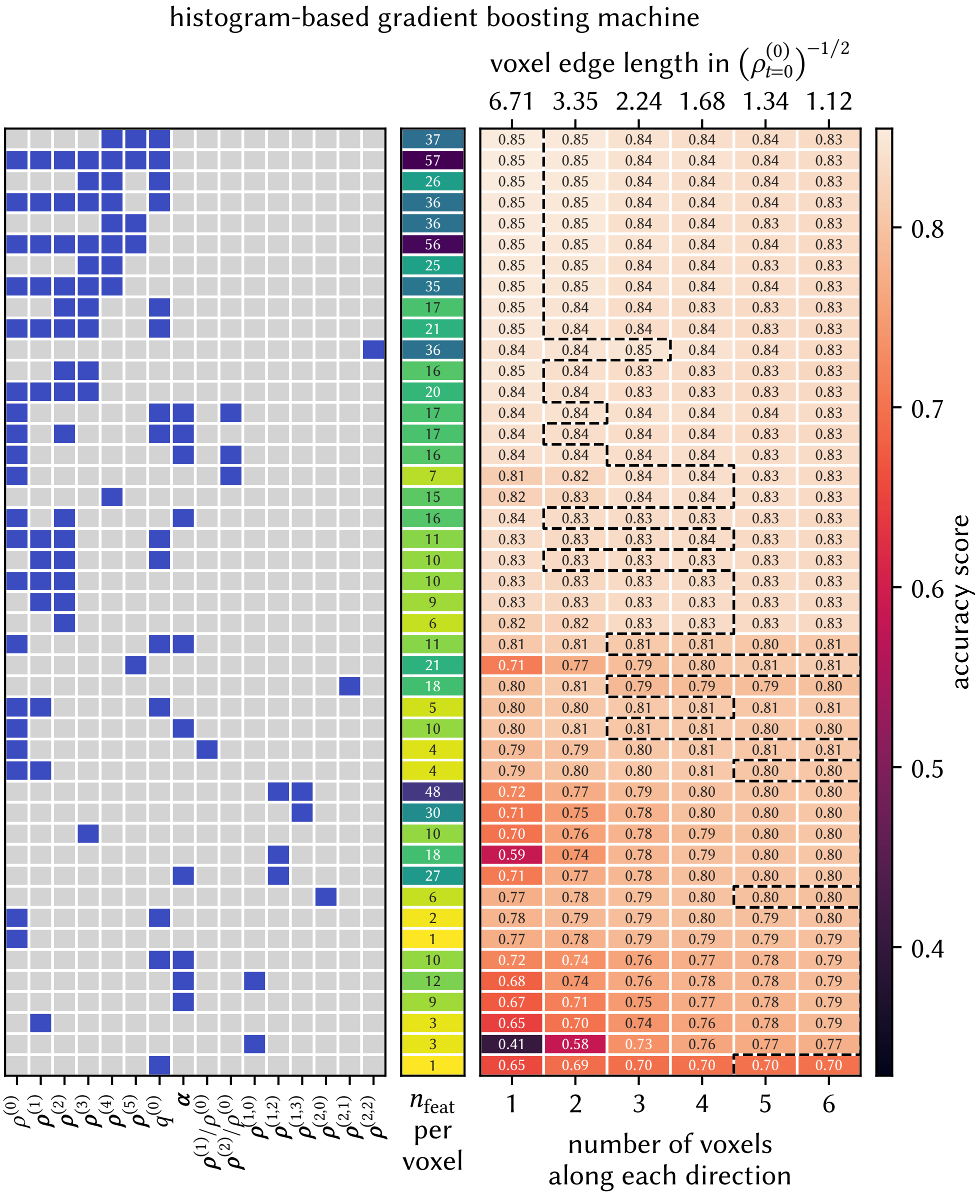


Figure 6.6.: Accuracy scores of the HGBM-based machine learning models for different field combinations and different spatial discretizations. Rows are ordered by the highest achieved by the respective field combination. The dashed line indicates the boundary for which an increase in spatial resolution results in an increased .

order CDD fields positively affects the quality of the model, with combinations having an even-ordered CDD field as their highest-ordered field being better than combinations having the next higher odd-ordered CDD field as their highest-ordered field. Taking all lower-order fields into account decreases the performance of the model slightly. Using the curvature density results in better model quality. Replacing the 1st-order CDD field $\boldsymbol{\rho}^{(1)}$ with the Kröner–Nye tensor $\boldsymbol{\alpha}$ makes the model worse. Combinations including only dislocation feature density tensors show hardly any improvement upon the baseline of 0.33. Finer spatial discretizations result in better accuracy scores.

Regardless of the spatial discretization, using the 1st-order CDD density field $\boldsymbol{\rho}^{(1)}$ or the Kröner–Nye tensor $\boldsymbol{\alpha}$ as sole fields results in accuracy scores of about 0.34. We observe the same for dislocation feature density tensors $\boldsymbol{\rho}^{(1,n)}$. This is about the baseline accuracy score of 0.33 for randomly classifying dislocation microstructures for this problem. Using the 3rd-order or 5th-order CDD density fields $\boldsymbol{\rho}^{(3)}$ and $\boldsymbol{\rho}^{(5)}$, respectively, as sole field results in accuracy scores ranging from 0.45 to 0.54. The field combinations $\{q^{(0)}\}$ and $\{q^{(0)}, \boldsymbol{\alpha}\}$ lead to accuracy scores between 0.63 to 0.67. All other field combinations reach an accuracy score of at least 0.73.

Histogram-based gradient boosting machine

For the models based on the HGBM algorithm, the single field performance ranked from worst to best is $\boldsymbol{\alpha}$, $\boldsymbol{\rho}^{(1)}$, $\boldsymbol{\rho}^{(3)}$, $\boldsymbol{\rho}^{(5)}$, $q^{(0)}$, $\rho^{(0)}$, $\boldsymbol{\rho}^{(2)}$, and $\boldsymbol{\rho}^{(4)}$. Overall, including higher-order CDD fields positively affects the quality of the model. Taking all lower-order fields into account decreases the performance of the model slightly. Using the curvature density results in better model quality. Replacing the 1st-order CDD field $\boldsymbol{\rho}^{(1)}$ with the Kröner–Nye tensor $\boldsymbol{\alpha}$ improves the model. Combinations including only dislocation feature density tensors do not perform as well as the ones including only dislocation density alignment tensors. Fine spatial discretizations are better for lower numbers of features per voxel, whereas high numbers of features result in coarser voxels being better.

With the exception of the field combinations that do not include an even-ordered CDD field, the performance of the models is somewhat stable with respect to the spatial discretization. Aforementioned exceptions show improvements in their accuracy score between 0.05 to 0.14. In contrast, the accuracy scores of other models change between 0.01 to 0.02.

Across algorithms

The highest accuracy scores achieved for each machine learning algorithm are

- 0.83 for GNB with the fields $\{\boldsymbol{\rho}^{(4)}, \boldsymbol{\rho}^{(5)}\}$ and 1 voxels along each direction,

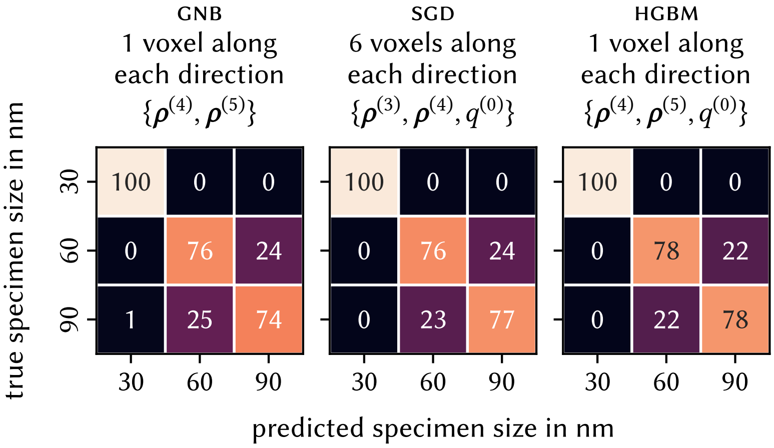


Figure 6.7.: Confusion matrices of the best performing field and discretization combinations of all used machine learning algorithms.

- 0.84 for SGD with the fields $\{\rho^{(3)}, \rho^{(4)}, q^{(0)}\}$ and 6 voxels along each direction, and
- 0.85 for HGBM with the fields $\{\rho^{(4)}, \rho^{(5)}, q^{(0)}\}$ and 1 voxel along each direction.

A trend observed across all tested algorithms is that using higher-ordered fields results in better accuracy in the predictions, especially for coarser spatial discretizations.

Models based on SGD or HGBM exhibit more stable behavior for different spatial discretizations. In contrast, models based on GNB exhibit drops in their accuracy score when increasing the spatial resolution from one to two voxels along each direction; between 0.02 to 0.05 for the field combinations that do not include the dislocation curvature density $q^{(0)}$, and 0.09 to 0.10 for field combinations that do include it.

6.2.2.3. Confusion matrices

The confusion matrices of aforementioned best performing machine learning models are shown in Figure 6.7. In all three cases, the dislocation microstructures of 30 nm specimen are correctly classified. The misclassification of dislocation microstructures taken from 60 nm specimen as belonging to 90 nm specimen and vice-versa is balanced.

Almost the same trend is seen for the best performing sequential CDD fields shown in Figure 6.8. But unlike the best performing SGD-based model, this

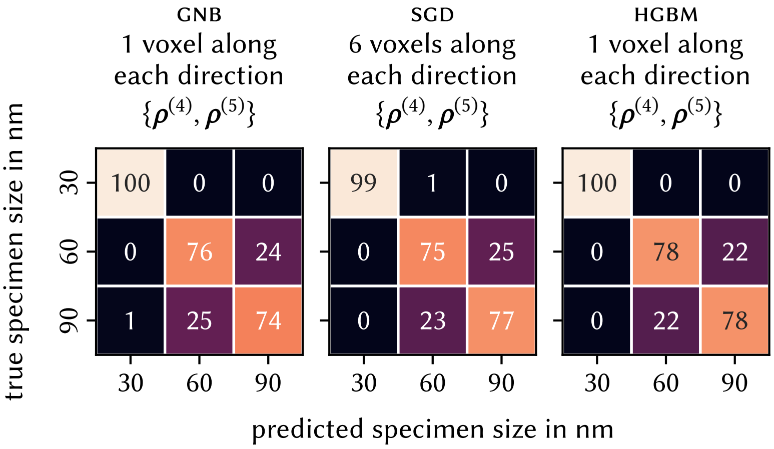


Figure 6.8.: Confusion matrices of the best performing sequential CDD field and discretization combinations of all used machine learning algorithms.

one does not perfectly classify microstructures from 30 nm specimen.

The aforementioned balance is not seen for all machine learning models trained with worst performing CDD sequential fields, as shown in Figure 6.9. In this case, the SGD-based machine learning model has a very strong preference of classifying dislocation microstructures taken from 60 nm and 90 nm specimen as belonging to 60 nm specimen. Although it does decrease the misclassification of microstructures of 60 nm specimen, it does so at the cost of misclassifying microstructures of 90 nm specimen in about 76 % of the cases. In weaker form, machine learning models based on the HGBM algorithm experience this trend.

The confusion matrices for feature sets $\{\rho^{(1)}, \rho^{(2)}\}$ and $\{\rho^{(3)}, \rho^{(4)}\}$ for 1 voxel along each direction are shown in Figure 6.10. As before, the correct classification of all dislocation microstructures from 30 nm specimen is observed. In the lower-order field combination, the machine learning model based on the GNB algorithm shows bias towards classifying the remaining microstructures as belonging to the 90 nm specimen. Contrary to that, the one based on the SGD algorithm exhibits strong bias towards classifying them as belonging to 60 nm specimen. No bias is observed for the machine learning model based on the HGBM. For the higher-order field combinations, the bias of the GNB-based machine learning model vanishes, and lessens for the ones based on SGD.

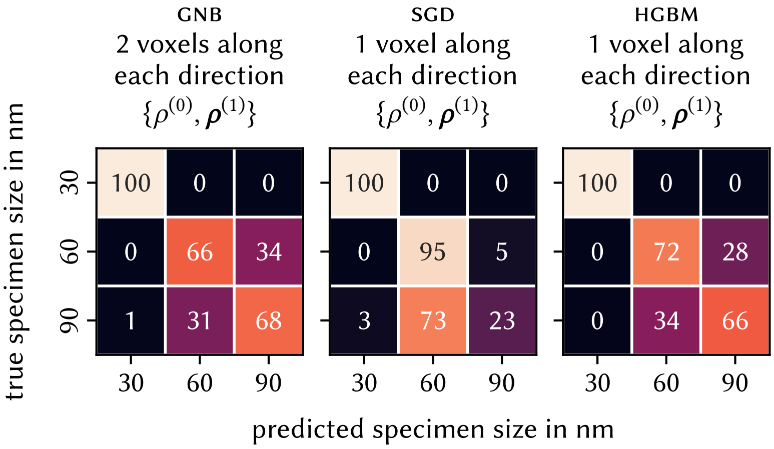


Figure 6.9.: Confusion matrices of the worst performing sequential CDD field and discretization combinations of all used machine learning algorithms.

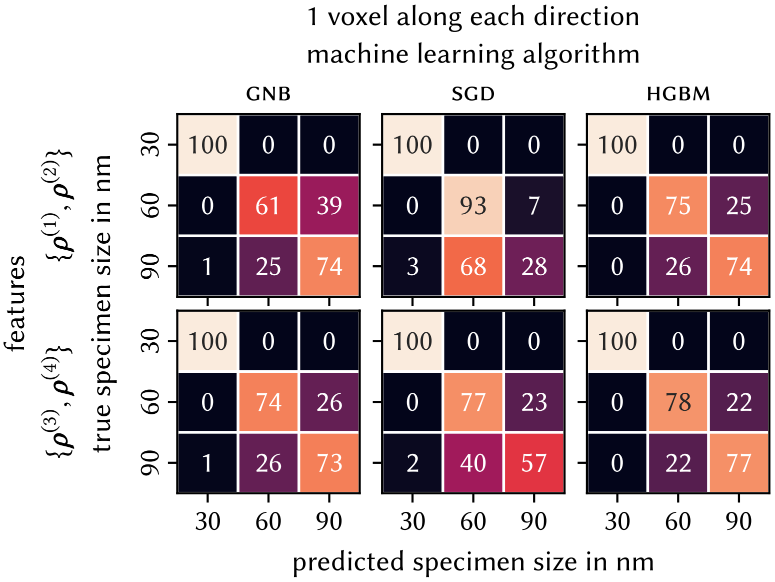


Figure 6.10.: Confusion matrices of the field combinations $\{\boldsymbol{\rho}^{(1)}, \boldsymbol{\rho}^{(2)}\}$ and $\{\boldsymbol{\rho}^{(3)}, \boldsymbol{\rho}^{(4)}\}$ for 1 voxel along each direction of all used machine learning algorithms.

6.2.2.4. Feature importance

The relative permutation feature importance for different field combinations, spatial discretizations and models are shown in Figures 6.11 to 6.19. Only the permutation feature importance for meshes with 1, 3 and 6 voxels along each direction are shown here. They exhibit the features observed for all discretizations. For a compilation of all spatial discretizations, see appendix A.

Gaussian naïve Bayes

In most field and spatial discretization combinations, the most important features within each machine learning model based on the GNB algorithm are based on the highest even-ordered dislocation density alignment tensor included. So when $\boldsymbol{\rho}^{(4)}$ is included, it will contribute most to the performance of the machine learning model, regardless of other fields that are included as well. If it is not included, but $\boldsymbol{\rho}^{(2)}$ is, features based on it are the most important ones – except if only one voxel is used and $\rho^{(0)}$ is included as well, in which case the latter is more important.

One notable exception to this rule are field combinations that include the dislocation curvature density $q^{(0)}$ and are discretized using more than 3 voxels along each axis. In these cases, features based on $q^{(0)}$ are the most important ones, albeit their importance diminishes with the addition of other fields. For lower resolutions of the spatial discretizations, the contribution of $q^{(0)}$ is negligible.

The orientation distributions contribute little to the performance of machine learning models including it in its field combination.

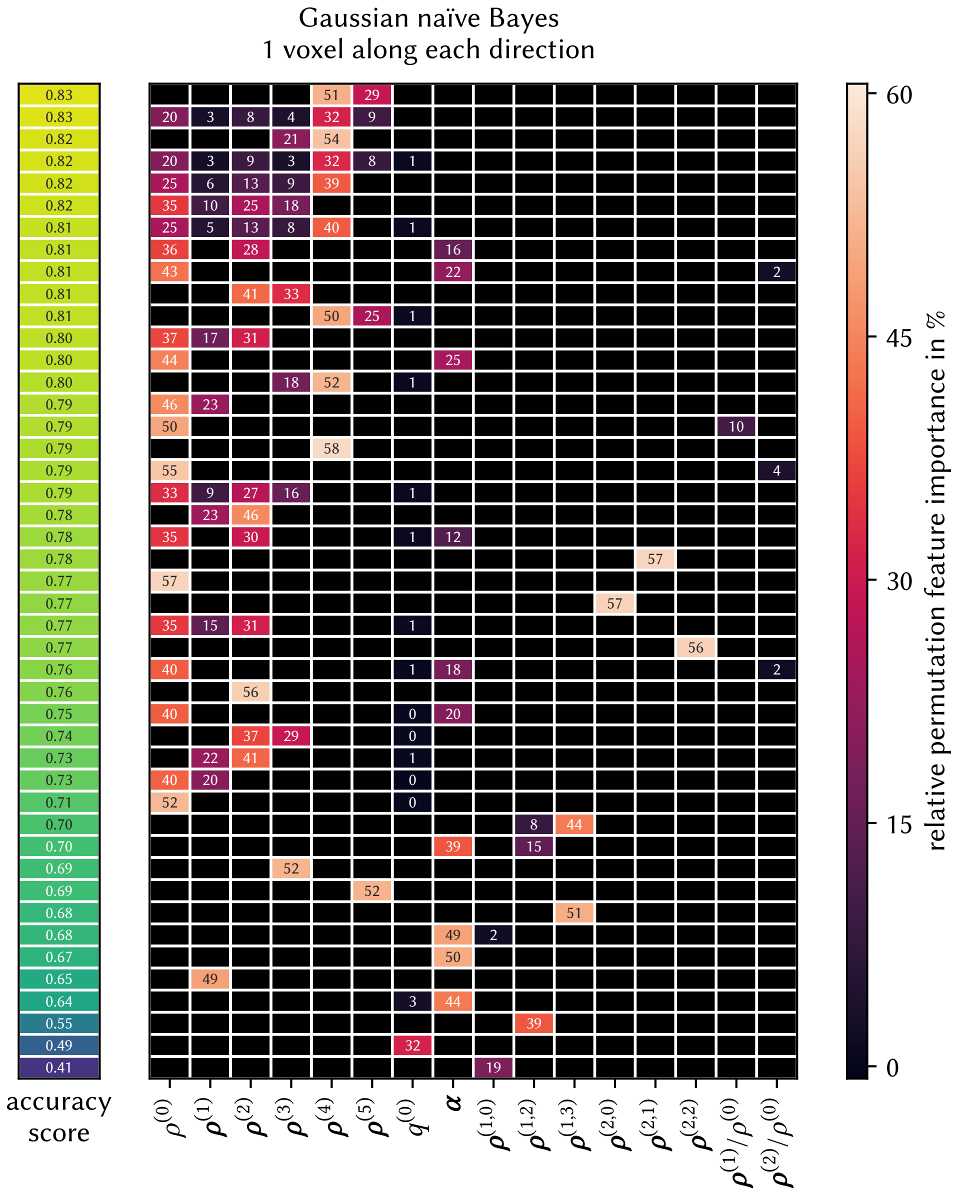
Stochastic gradient descent

The features contributing most to the performance of machine learning models based on the SGD algorithm are even-ordered dislocation density alignment tensors, irrespective of the spatial discretization. If several are included in the field combination, the higher-ordered ones are more important.

The only other features that contribute in a significant manner are the dislocation curvature density $q^{(0)}$ for coarse spatial discretizations, and the 2nd-order dislocation orientation distribution $\boldsymbol{\rho}^{(2)}/\rho^{(0)}$ for higher spatial resolutions.

Histogram-based gradient boosting machine

Irrespective of other fields included in the field combination or the spatial discretization, the total dislocation density $\rho^{(0)}$ constitutes the most important features for machine learning model based on the HGBM algorithm. If other even-ordered dislocation density alignment tensors are included, features



accuracy score	$\rho^{(0)}$	$\boldsymbol{\rho}^{(1)}$	$\boldsymbol{\rho}^{(2)}$	$\boldsymbol{\rho}^{(3)}$	$\boldsymbol{\rho}^{(4)}$	$\boldsymbol{\rho}^{(5)}$	$q^{(0)}$	$\boldsymbol{\alpha}$	$\boldsymbol{\rho}^{(1,0)}$	$\boldsymbol{\rho}^{(1,2)}$	$\boldsymbol{\rho}^{(1,3)}$	$\boldsymbol{\rho}^{(2,0)}$	$\boldsymbol{\rho}^{(2,1)}$	$\boldsymbol{\rho}^{(2,2)}$	$\boldsymbol{\rho}^{(1)}/\rho^{(0)}$	$\boldsymbol{\rho}^{(2)}/\rho^{(0)}$
0.83					51	29										
0.83	20	3	8	4	32	9										
0.82				21	54											
0.82	20	3	9	3	32	8	1									
0.82	25	6	13	9	39											
0.82	35	10	25	18												
0.81	25	5	13	8	40		1									
0.81	36		28					16								
0.81	43							22								2
0.81			41	33												
0.81					50	25	1									
0.80	37	17	31													
0.80	44							25								
0.80				18	52		1									
0.79	46	23														
0.79	50														10	
0.79					58											
0.79	55															4
0.79	33	9	27	16			1									
0.78		23	46													
0.78	35		30				1	12								
0.78													57			
0.77	57															
0.77												57				
0.77	35	15	31				1									
0.77														56		
0.76	40						1	18								2
0.76			56													
0.75	40						0	20								
0.74			37	29			0									
0.73		22	41				1									
0.73	40	20					0									
0.71	52						0									
0.70										8	44					
0.70								39		15						
0.69				52												
0.69						52										
0.68											51					
0.68								49	2							
0.67								50								
0.65		49														
0.64							3	44								
0.55										39						
0.49							32									
0.41									19							

Figure 6.11.: Relative permutation feature importance of each employed field for all field combinations for 1 voxel along each direction for the GNB-based machine learning models. The field combinations are sorted by the achieved accuracy score.

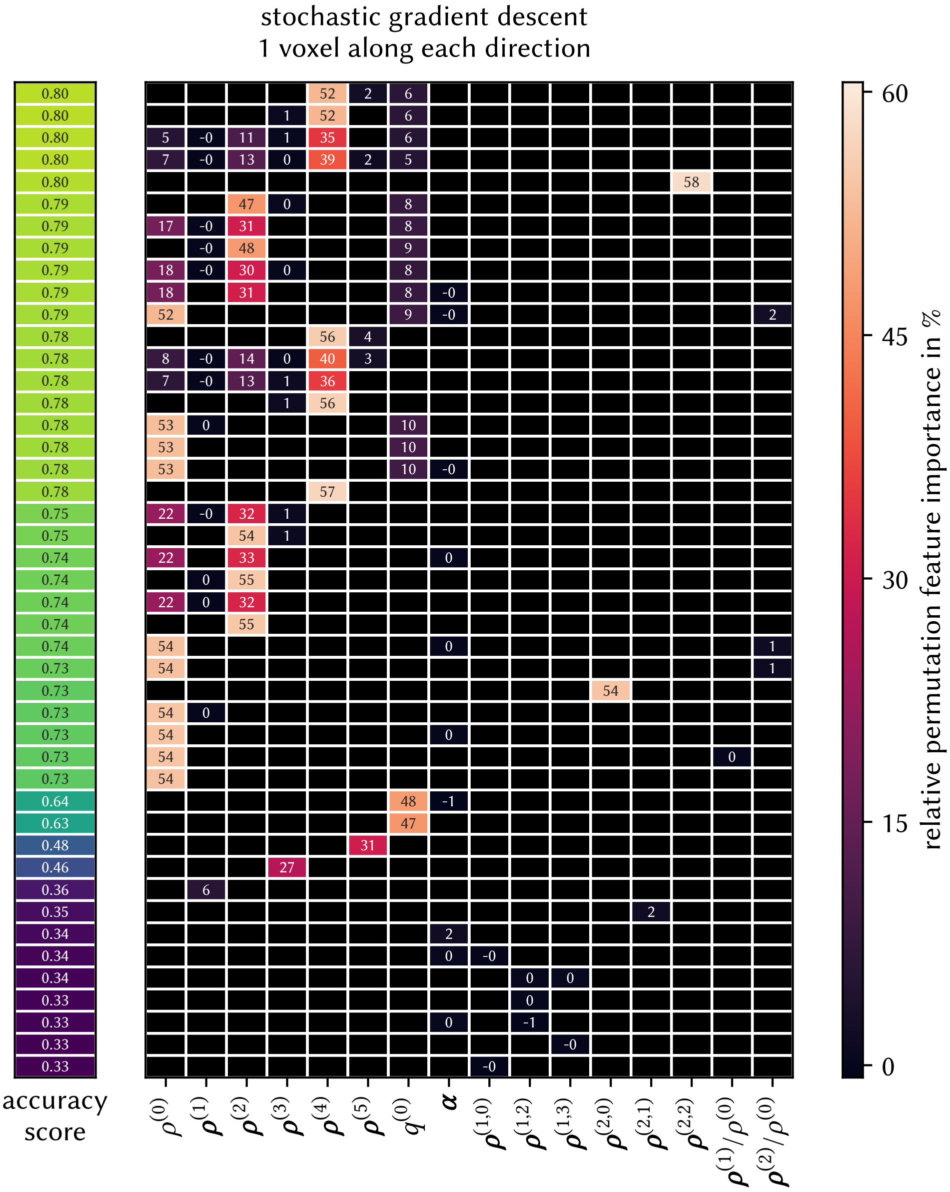


Figure 6.12.: Relative permutation feature importance of each employed field for all field combinations for 1 voxel along each direction for the SGD-based machine learning models. The field combinations are sorted by the achieved accuracy score.

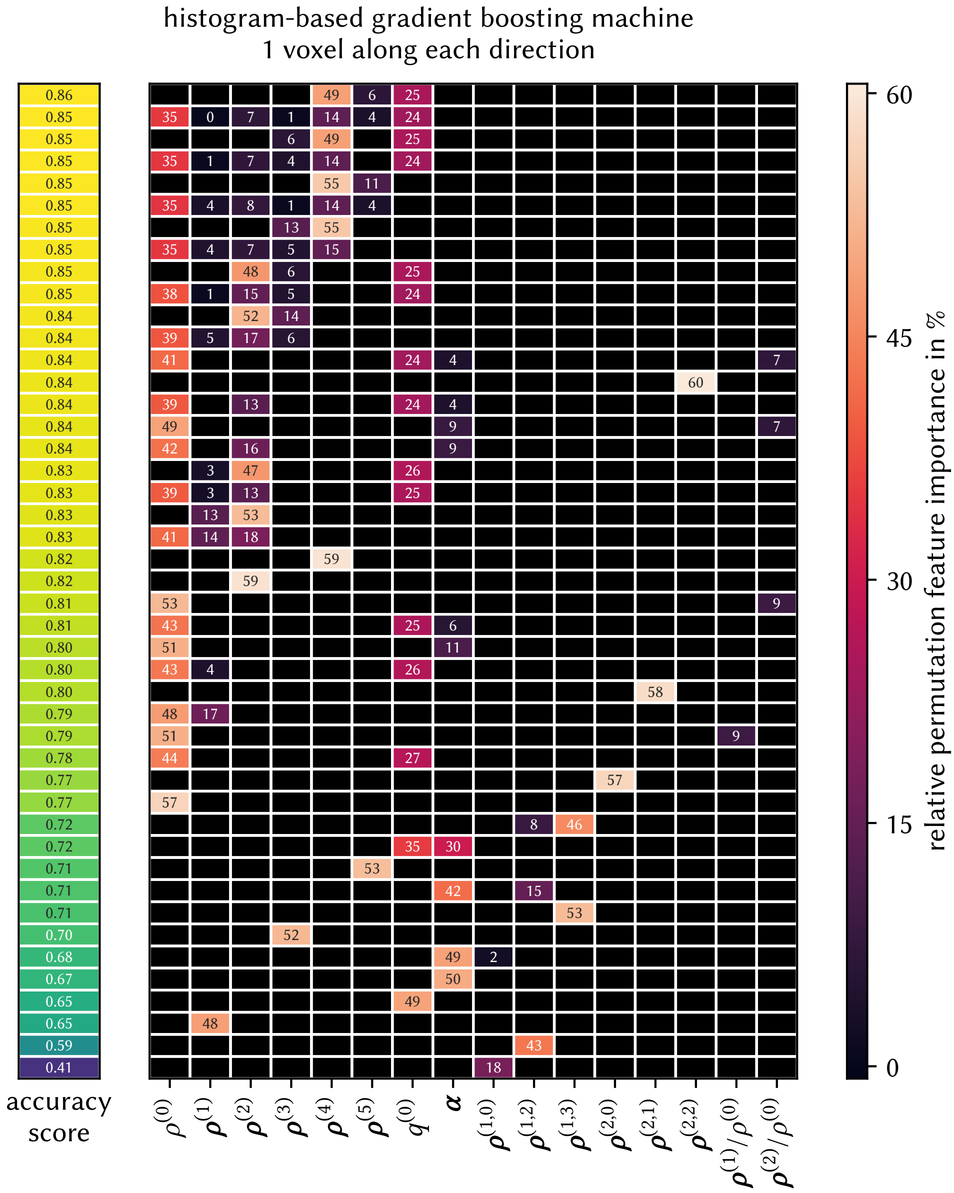


Figure 6.13.: Relative permutation feature importance of each employed field for all field combinations for 1 voxel along each direction for the HGBM-based machine learning models. The field combinations are sorted by the achieved accuracy score.

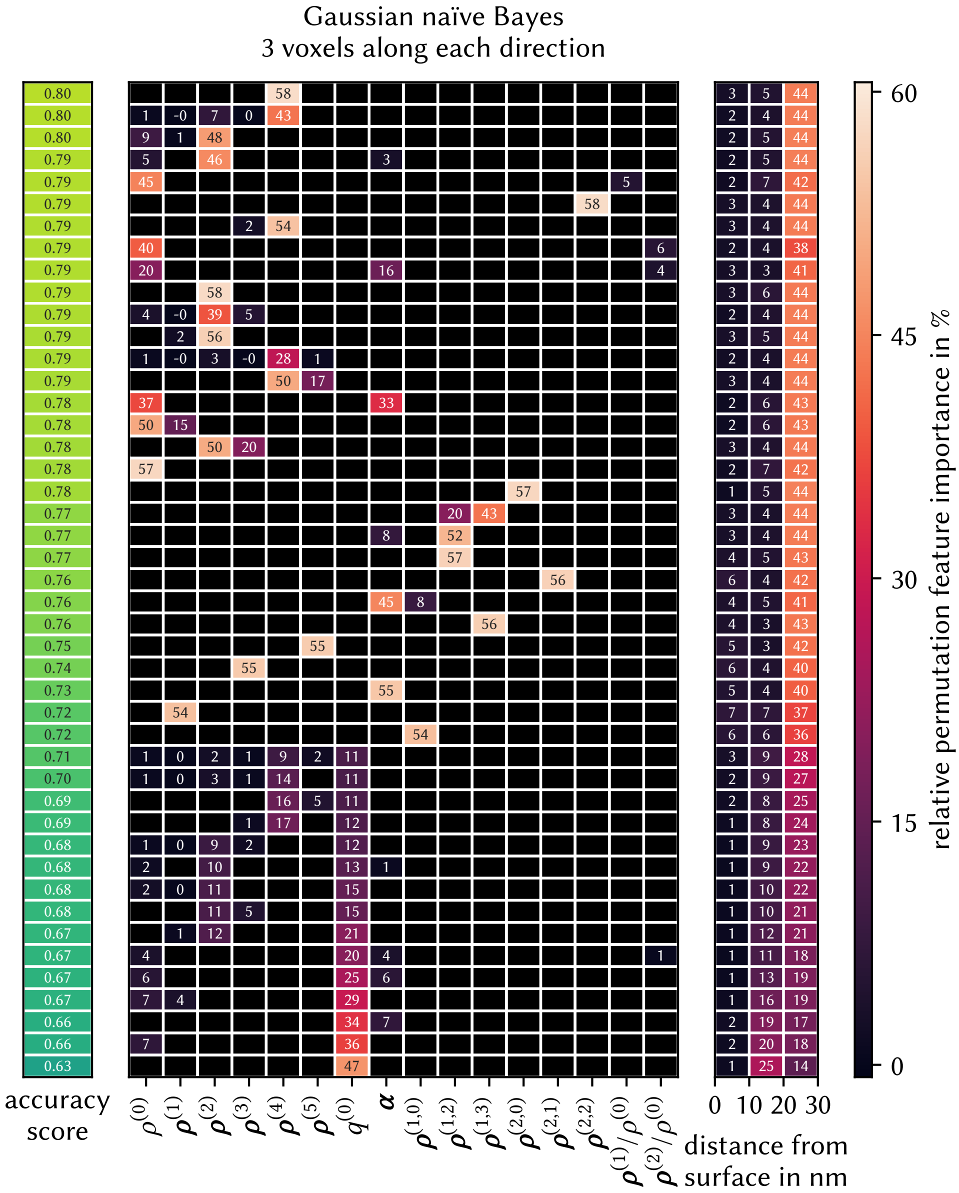


Figure 6.14.: Relative permutation feature importance of each employed field and each distance from the surface of each voxel for all field combinations for 3 voxels along each direction for the GNB-based machine learning models. The field combinations are sorted by the achieved accuracy score.

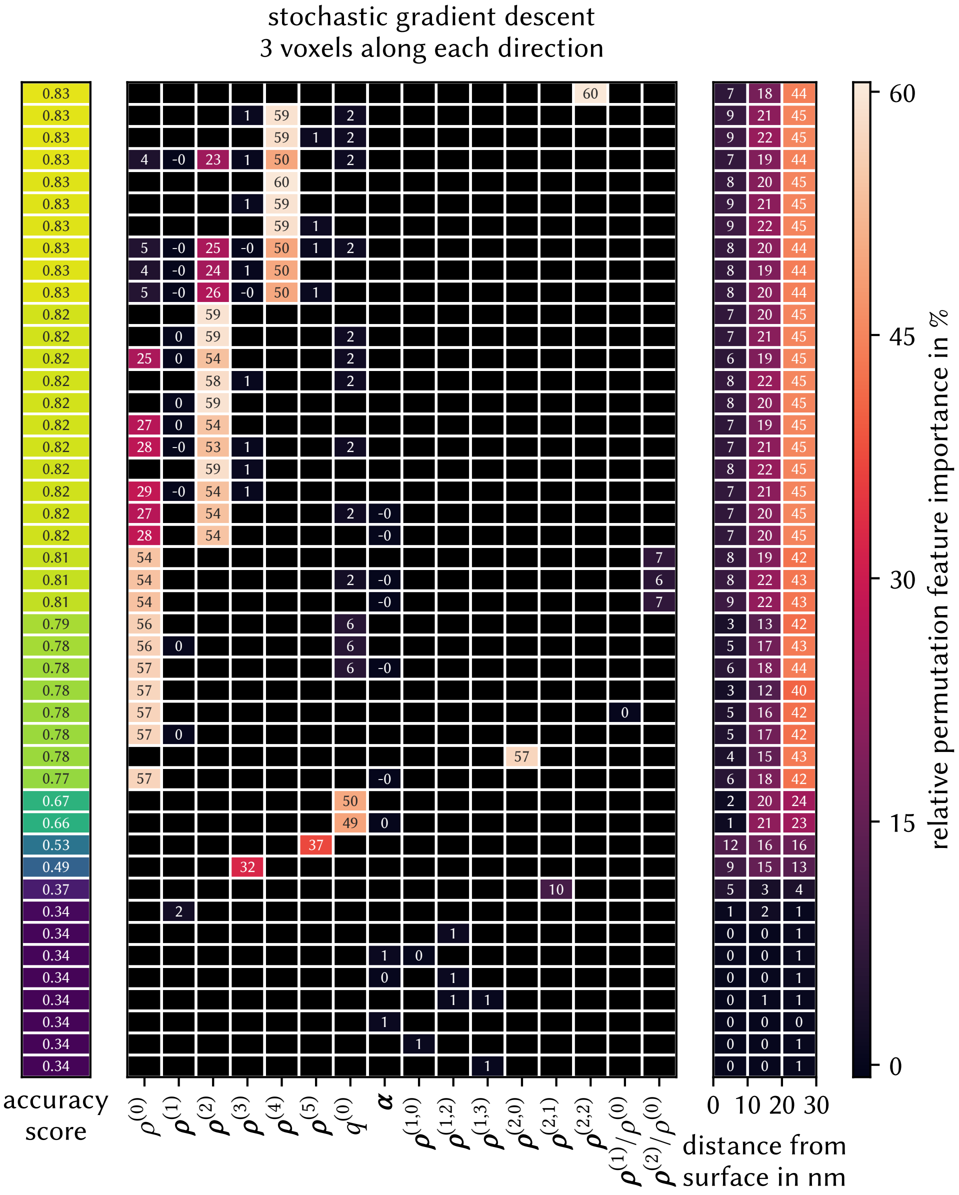


Figure 6.15.: Relative permutation feature importance of each employed field and each distance from the surface of each voxel for all field combinations for 3 voxels along each direction for the SGD-based machine learning models. The field combinations are sorted by the achieved accuracy score.

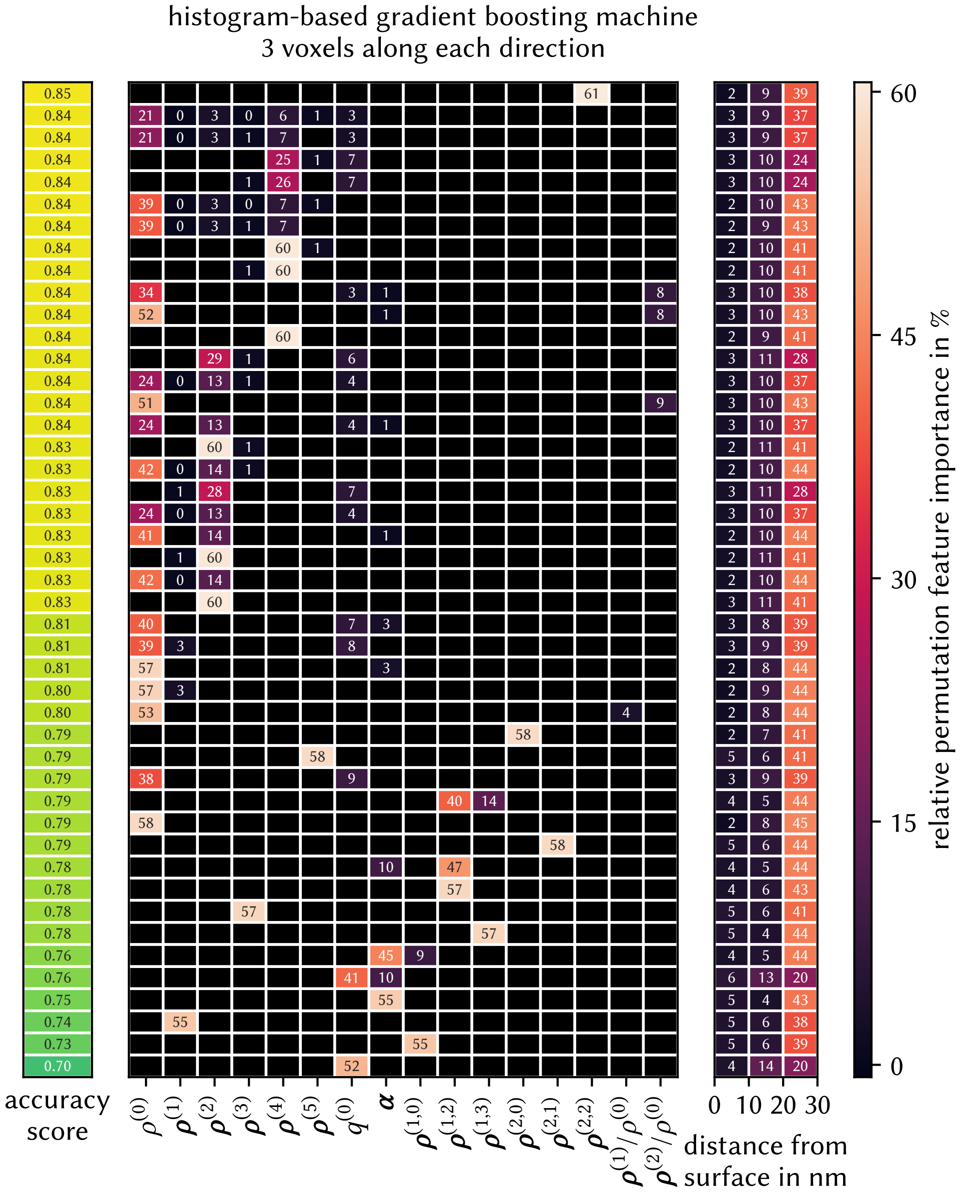


Figure 6.16.: Relative permutation feature importance of each employed field and each distance from the surface of each voxel for all field combinations for 3 voxels along each direction for the HGBM-based machine learning models. The field combinations are sorted by the achieved accuracy score.

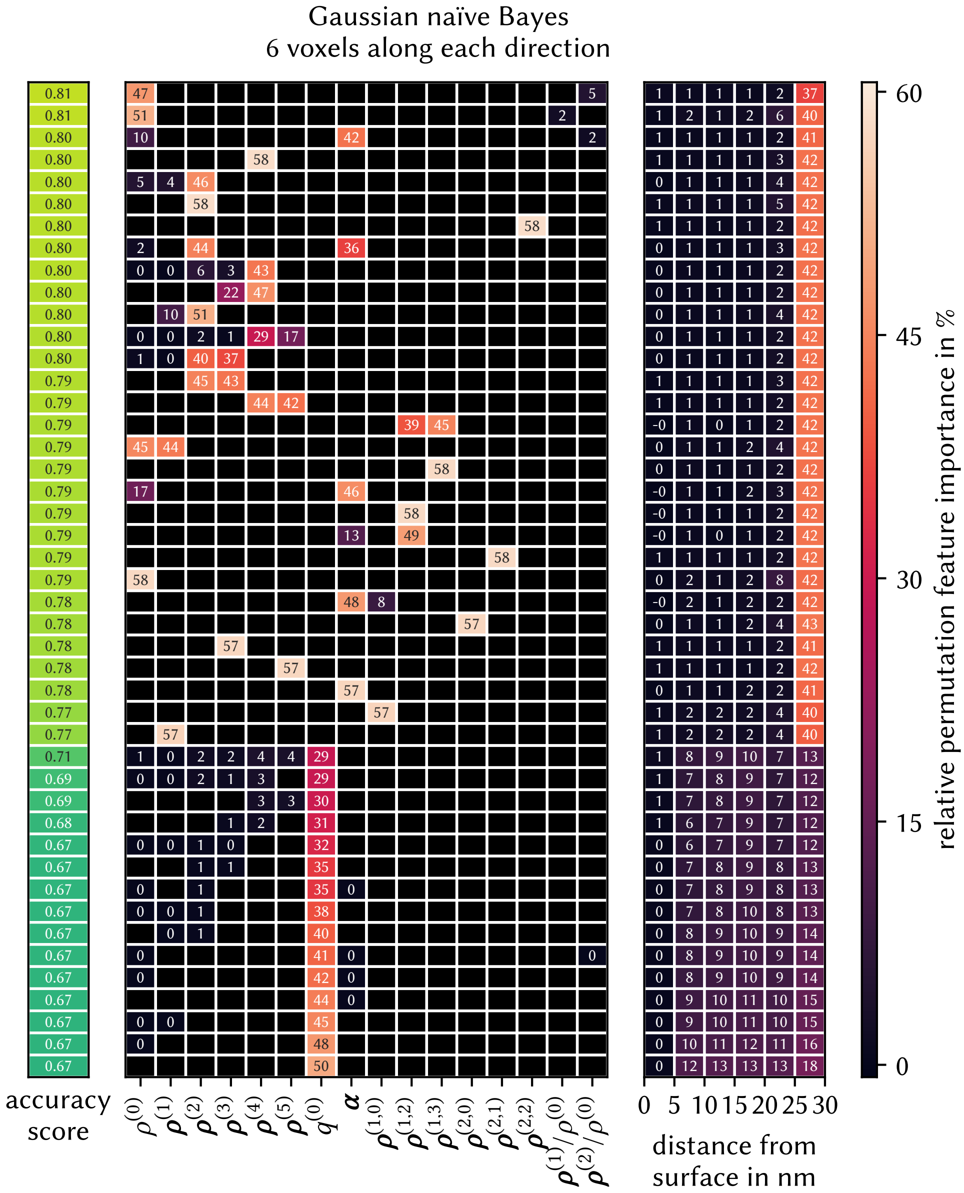


Figure 6.17.: Relative permutation feature importance of each employed field and each distance from the surface of each voxel for all field combinations for 6 voxels along each direction for the GNB-based machine learning models. The field combinations are sorted by the achieved accuracy score.

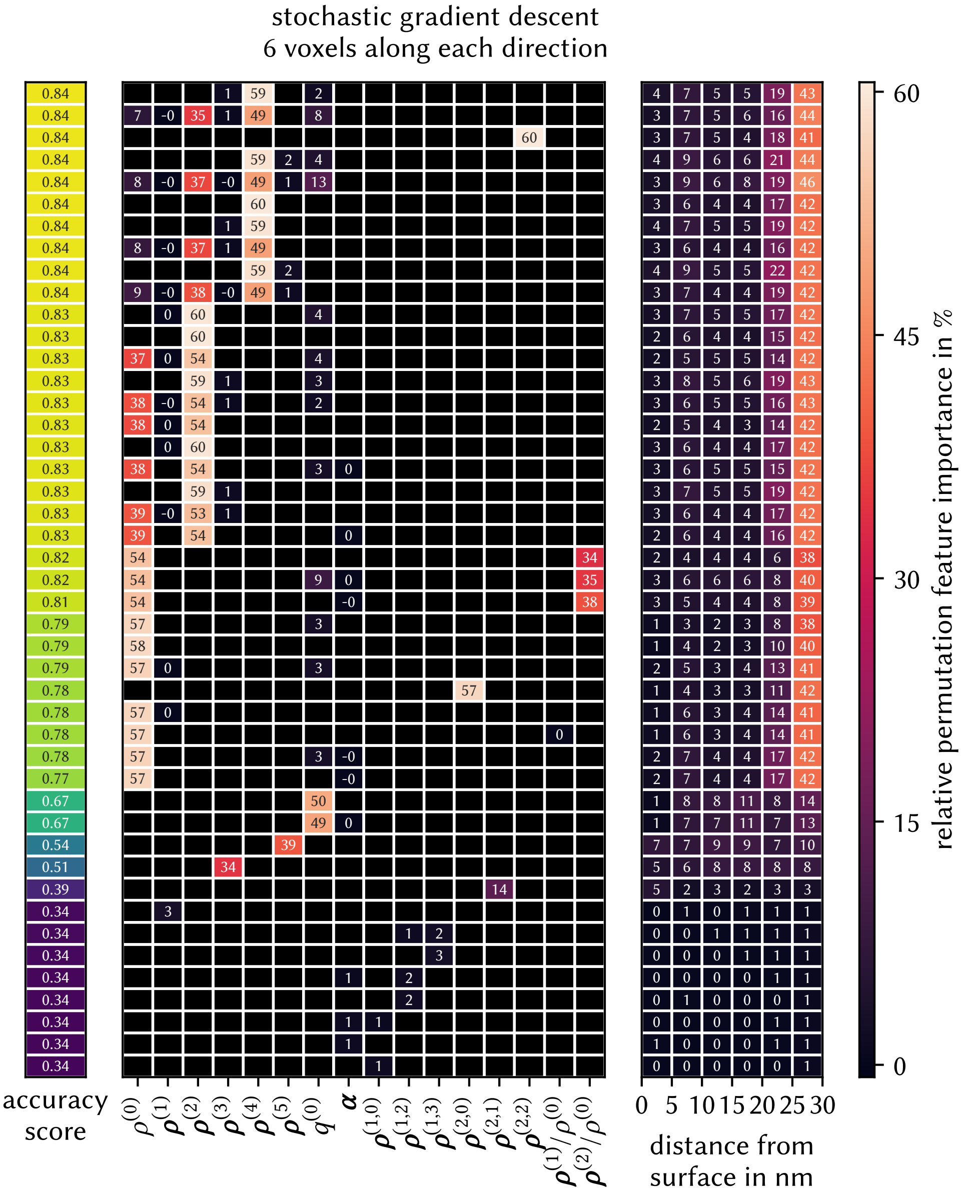


Figure 6.18.: Relative permutation feature importance of each employed field and each distance from the surface of each voxel for all field combinations for 6 voxels along each direction for the SGD-based machine learning models. The field combinations are sorted by the achieved accuracy score.

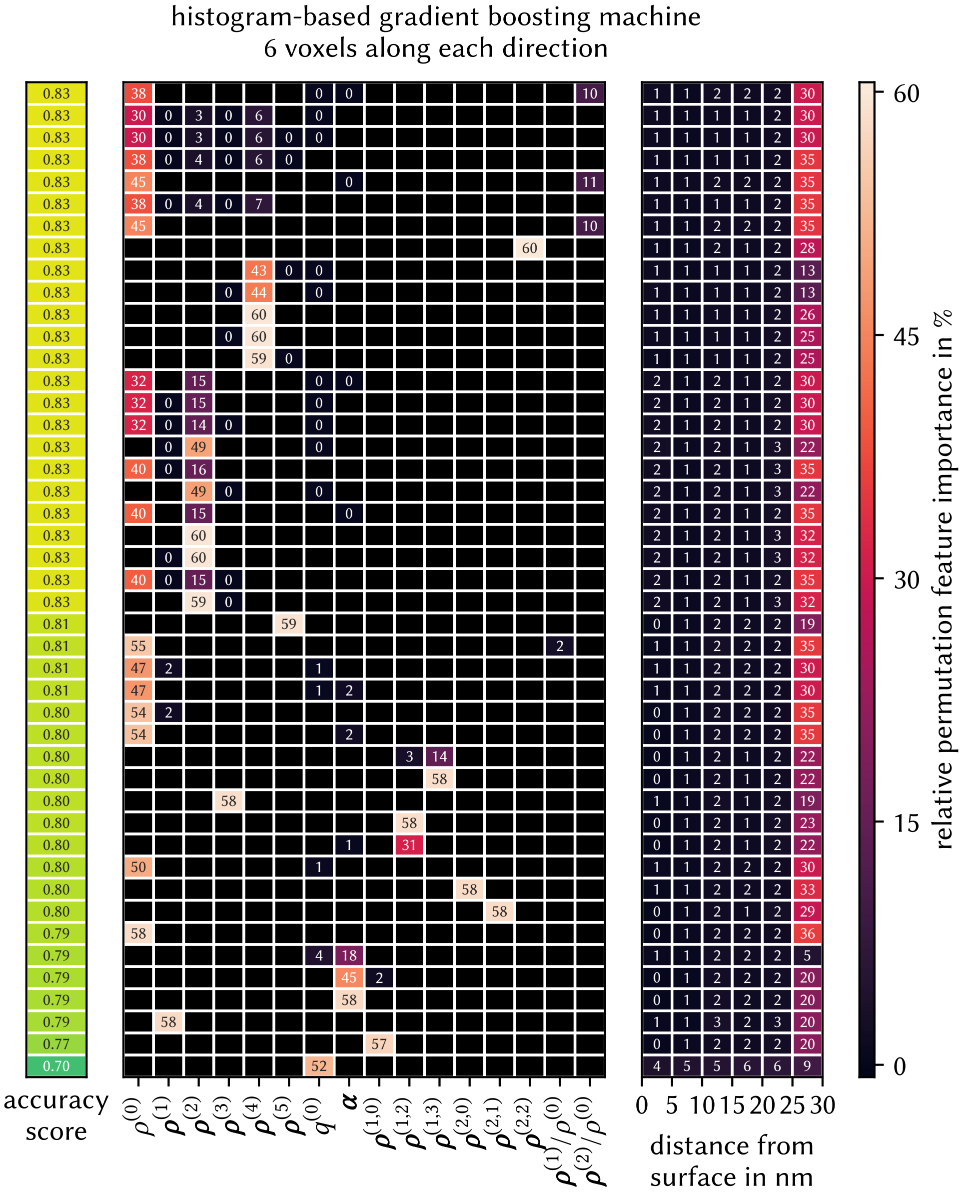


Figure 6.19.: Relative permutation feature importance of each employed field and each distance from the surface of each voxel for all field combinations for 6 voxels along each direction for the HGBM-based machine learning models. The field combinations are sorted by the achieved accuracy score.

based on $\boldsymbol{\rho}^{(4)}$ are more important than $\boldsymbol{\rho}^{(2)}$.

For spatial resolutions lower than 3 voxels along each direction, the features based on the dislocation curvature density $q^{(0)}$ also contribute to the performance of the machine learning models.

Location of imporant features

Universally across all machine learning model, the most important features are the ones with maximum distance from the free surface. For the SGD-based machine learning model, features whose distance is the second furthest from the free surface also contribute, albeit not as much as the former features. This effect is less pronounced for machine learning models

1. based on the GNB algorithm that contain the dislocation curvature density $q^{(0)}$,

2. based on the SGD algorithm that do not contain an even-ordered dislocation density alignment tensor $\rho^{(0)}$, $\boldsymbol{\rho}^{(2)}$, or $\boldsymbol{\rho}^{(4)}$,

3. based on the HGBM algorithm that contain the dislocation curvature density $q^{(0)}$ or do not contain an even-ordered dislocation density alignment tensor $\rho^{(0)}$, $\boldsymbol{\rho}^{(2)}$, or $\boldsymbol{\rho}^{(4)}$.

6.2.2.5. Learning curves

Learning curves for field combinations $\{\boldsymbol{\rho}^{(1)}, \boldsymbol{\rho}^{(2)}\}$ and $\{\boldsymbol{\rho}^{(3)}, \boldsymbol{\rho}^{(4)}\}$ for 1 voxel along each direction are shown in figures 6.20 and 6.21, respectively. For the field combination $\{\boldsymbol{\rho}^{(1)}, \boldsymbol{\rho}^{(2)}\}$, GNB plateaus after 12.5 % of the available training observations are used, i.e., 39 370 observations, and SGD plateau after 25 %, i.e., 78 739 observations. In contrast, HGBM improves up to the total amount of training observations.

For the field combination $\{\boldsymbol{\rho}^{(3)}, \boldsymbol{\rho}^{(4)}\}$, GNB plateaus for more than 25 %, whereas GNB does so for more than 12.5 %. For few observations, HGBM performs comparable to SGD, but improves until all observations are used for training.

The learning curve for the latter field combination is also shown for 6 voxels along each direction in Figure 6.22. The performance of GNB plateaus again for more than 25 %. Both, SGD and HGBM improve up to using all observations.

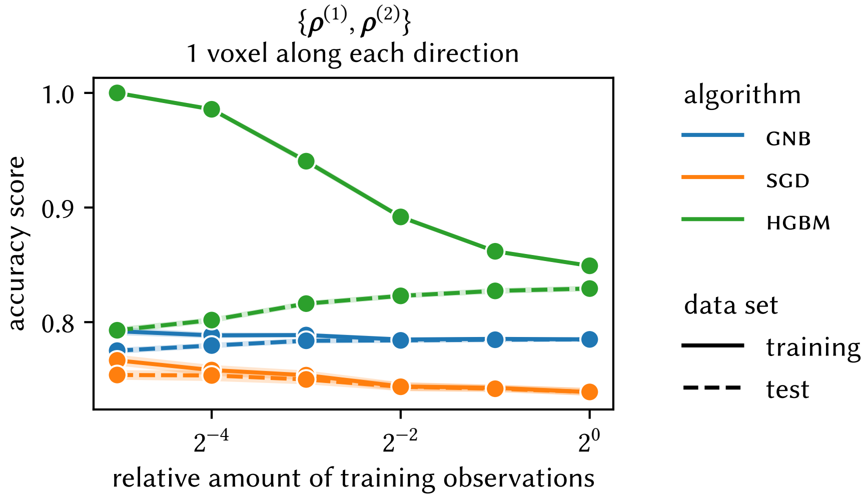


Figure 6.20.: Learning curve for the field combination $\{\rho^{(1)}, \rho^{(2)}\}$ and 1 voxel along each direction.

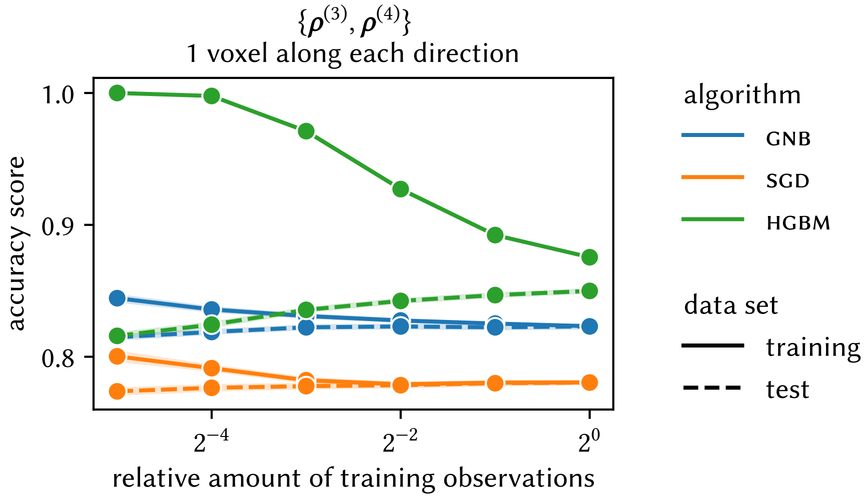


Figure 6.21.: Learning curve for the field combination $\{\rho^{(3)}, \rho^{(4)}\}$ and 1 voxel along each direction.

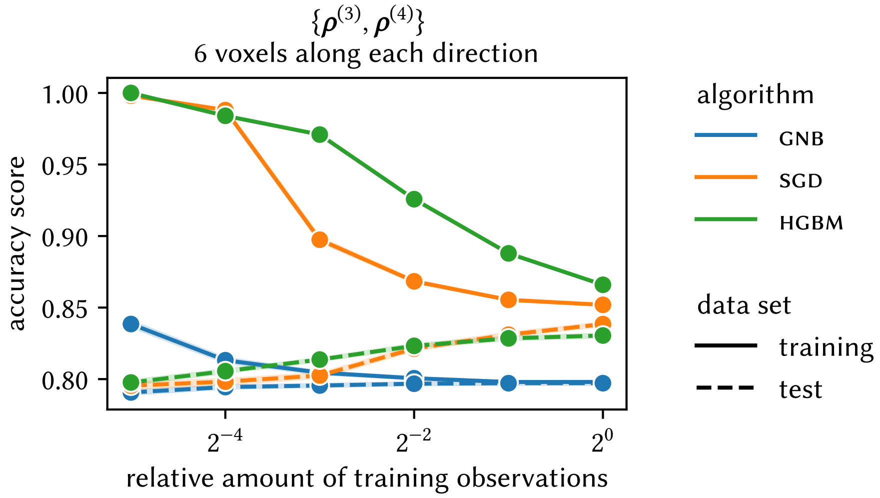


Figure 6.22.: Learning curve for the field combination $\{\rho^{(3)}, \rho^{(4)}\}$ and 6 voxels along each direction.

6.2.3. Discussion

6.2.3.1. Average dislocation density

The average dislocation densities nicely highlight the important role of the dislocation-surface interaction in nanoparticles. From section 2.2.4, we know that dislocations are attracted by open surfaces. The closer the surface, the stronger the attractive forces. And in a nanoparticle, all surfaces are more or less close. Additionally, dislocations are able to leave through open surfaces. In combination, this leads to the gradients we observe in the average dislocation density for all specimen sizes.

The average dislocation density of the 30 nm sample differs significantly from the ones of larger specimen for these reasons. Every side of the samples of the 30 nm specimen are open surfaces, so the dislocation density close to all faces of the samples exhibit a strong drop in density compared to the center. High densities in the center of the sample observed for higher resolutions support this. At the center, the attractive forces of the open surfaces acting on the dislocations cancel each other out and enable the dislocations to be concentrated there on average.

Open surfaces that are not part of the sample also influence the dislocation microstructure observed within it. We can see this when comparing the average total dislocation density of the 60 nm and 90 nm specimen. They share the gradient in the density towards the open surface of the sample. Along the perpendicular directions, i.e., along the y-axis and z-axis, samples taken from the 60 nm samples show a decreasing density value for larger distances from the center x-axis. This is not observed to the same extent in samples of the 90 nmspecimen. Density values furthest from the open surface in the center of both y-axis and z-axis are also larger in the samples of the 60 nm specimen compared to the ones of the 90 nm specimen. The reason for this is that this location is the center of the 60 nm specimen. As mentioned before for the 30 nm specimen, dislocations at the center do not feel attractive forces of the surfaces, as they cancel each other out. On average this leads to a higher likelihood of finding dislocations there, which results in the average total dislocation density being highest there. This same balance of forces is only found outside the samples taken from 90 nm. Because of this, the maximum total dislocation density observed from 60 nm specimen is higher than that of 90 nm.

But, if we compare the differences of the total dislocation density between samples taken from 60 nm and 90 nm specimen to the differences of 30 nm specimen samples with the former, they are very small.

6.2.3.2. High success of classifying 30 nm compared to larger specimen

These large differences in their average total dislocation density are a first hint at why the correct classification of dislocation microstructures exhibits such a high accuracy rate. But, the machine learning models classify single observations instead of the averages we have discussed before. The accuracy with which dislocation microstructures of 30 nm specimen are classified correctly therefore implies that the surrounding open surfaces of the samples also impact the individual microstructures to a large extent. The probability density functions for the total dislocation density $\rho^{(0)}$ for a spatial discretization of 1 voxel along each direction are shown in Figure 6.23. We can see that the distribution of obtained from 30 nm specimen has little overlap with the distributions of larger specimen. Thus, based on the total dislocation density of the entire sample alone, a separation of 30 nm specimen against the others is feasible. Additional information is required for correctly distinguishing between dislocation microstructures obtained from 60 nm and 90 nm specimen. This is evident from the accuracy score increasing with the addition of more/other fields.

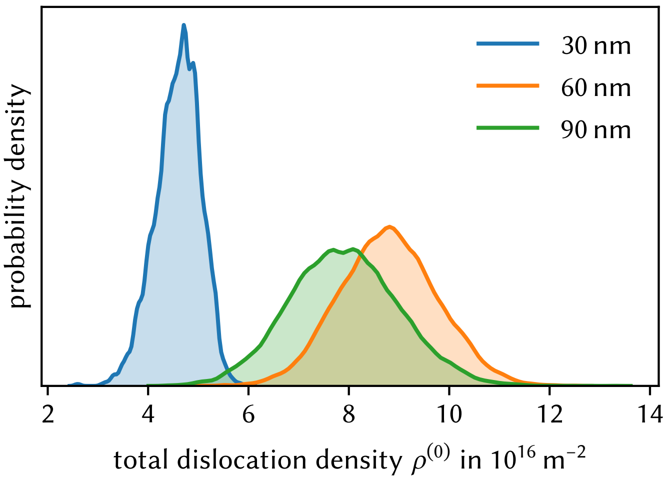


Figure 6.23.: Probability density functions of the total dislocation density $\rho^{(0)}$ for a spatial discretization of 1 voxel along each direction.

6.2.3.3. Importance of features further away from the common surface

From the relative feature importances shown in section 6.2.2.4, we may conclude that the most important information is found opposite of the common open surface. As outlined in section 6.2.3.1, the largest differences in the average total dislocation density are also found towards this direction. From the feature importances results, we conclude that this is not limited to the average total dislocation density, but to single observations and other features as well.

When we consider that the one thing samples of all specimen sizes have in common is the open surface in positive x-direction, it makes sense that the dislocation microstructures close to it are most similar there. As the distance to this common factor increases, so do the differences in the microstructure. And it is these differences that the machine learning models learn for the classification of the specimen size based on the dislocation microstructure.

6.2.3.4. Impact of even-order CDD fields

Regardless of the spatial discretization or the machine learning algorithm we base our machine learning models on, even-order CDD fields exhibit far higher feature importance than odd-order CDD fields. There are two potential reasons for this.

The first reason is that we consider only the global dislocation density alignment tensors, and not the slip system specific ones. This means that only the sense of the dislocations within a subvolume are averaged, irrespective of their Burgers vector. Thus, the deformation character of the dislocations is neglected. However, if this were the case we would expect the Kröner–Nye tensor and the dislocation feature density tensors to generally perform better than the odd-order CDD fields. As this is not the case, we would not expect any benefits from extracting slip system specific odd-order dislocation density alignment tensor fields.

The other reason is that the dislocation microstructure comprises mostly SSDs. As initial dislocation microstructure, we used dipolar edge dislocation loops that were randomly placed and whose slip systems were uniformly sampled from all possible slip systems available in the material. During the subsequent relaxation, no external load is applied. Hence, there is no driving force that would lead to deformation of the specimen, and therefore no formation of GND microstructures takes place in a characteristic manner, only locally due to fluctuations in the random initial conditions. For the machine learning models, features that originate from the odd-order dislocation density alignment tensors that are associated with GNDs are therefore only noise that contains nothing to learn from. This is why the even-order dislocation density alignment tensors exhibit a far higher feature importance than the odd-order ones.

6.2.3.5. Implications of simplifications in the simulations

The setup we used for the relaxation of dislocation microstructures is not entirely realistic since we neglected the formation of junctions and cross-slip. In section 5.3.1 we argued that there is a significant difference in the dislocation microstructure that we obtain depending on whether we consider cross-slip or not. We would expect the gradients in the dislocation density to be less pronounced towards the open surface. But as we also observed in section 5.3.3, the dislocation microstructures that do not consider cross-slip vary more than the ones that do consider cross-slip. We might therefore even argue that the classification problem was harder than it would be under more realistic conditions. However, the previous observations required the dislocation microstructures to evolve under tensile load first for the effects to become pronounced. As we only relaxed a random initial dislocation microstructure, we would like to argue that the effect of not considering aforementioned dislocation reactions does not weaken our argument that dislocation density fields are sufficient as input variables for machine learning models.

6.3. Indentation

Similarly to the relaxation experiment, we want to classify the size of a nanoparticle from the dislocation microstructure resulting from indentation. In addition to surface interactions, the stress field due to the indenter load affects the evolution of dislocation microstructures during nanoparticle indentation experiments. We also increase the number of particle sizes to make the classification more similar to a regression problem.

6.3.1. Data generation

The data for the indentation experiment was generated in four steps. Using the FEM software Abaqus FEA [76], stress fields within the particles due to the indenter were computed. They were then used within the microMegas DDD code [47] to simulate the indentation and evolve the dislocation microstructure. The setup of the simulations was done in collaboration with Shyamal Roy. Using the D2C method, we then extracted the dislocation microstructure features. Subsequently, we trained machine learning models using these features and evaluated their performance. In the following, we outline the last three steps in more details.

6.3.1.1. Generation and evolution of dislocation microstructures

We follow the method used by Roy et al. [77] to generate dislocation microstructures that form during nanoindentation of gold nanoparticles on a rigid sapphire substrate. This method is briefly outlined in the following, for a more detailed description the reader is referred to this article.

The geometry of the particles is based on the Winterbottom construction [78]. Based on the free surface energies of gold and the interface energy of gold with sapphire, the resulting particle shape shown in Figure 6.24 minimizes the total energy. Both, the substrate interface and the opposite surface that is indented are the crystallographic (1 1 1) planes. The stress field resulting from the indenter was determined via the FEM, where the indenter was simulated via a rigid sphere in a displacement-controlled manner. Elastic anisotropy was considered. Zero displacement boundary conditions were imposed at the substrate interface to imitate a rigid substrate. A rigid, spherical indenter with a radius of 5.5 nm was used to load the particle. The final stress field was then superimposed within the DDD simulations. This is an approximation of the real contact problem that we use for computational efficiency reasons, as we want to generate a large number of realizations for the training of the machine

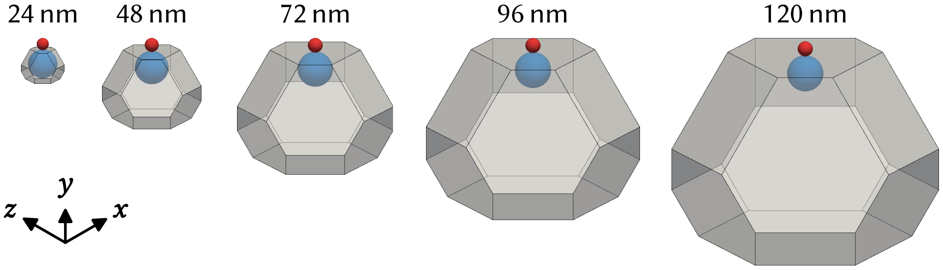


Figure 6.24.: Specimen (gray), indenter (red) and the domains of the samples (blue) whose dislocation microstructure are considered for the machine learning models.

learning models. During these simulations, dislocations were allowed to leave the particle on all sides except the substrate interface, which was modeled as impenetrable. The shape of the particles was kept constant during the simulations, i.e., dislocations did not alter the surface. Image forces were only considered for dislocations intersecting the free surfaces in a simplified manner, following the procedure proposed by [79, 80]. Thus, the overall driving forces for the microstructure evolution are the superimposed indenter stress field obtained via the FEM, approximate image forces due to the free surfaces, and internal dislocation-dislocation interactions.

The nucleation of dislocations was accounted for by introducing circular dislocation loops with radius 1.5 nm beneath the indenter. Dislocation loops of this size are only able to expand when the local resolved shear stress experienced by them exceeded 1.7 GPa, which therefore is the lower stress limit. At each time step, at most 5000 attempts were made to introduce a dislocation loop on a random slip system in random positions within a cube-shaped subvolume beneath the indenter with an edge length of 15 nm. In each attempt, a dislocation loop was preemptively introduced and its Peach–Köhler forces determined. If at least 75 % of the dislocation line segments would expand, this dislocation is actually introduced to the system and no further attempts are made. At least 0.3 ns wait time were used between a successful nucleation and the next attempts.

As the dislocations expand, plastic strain accumulates. We prescribed a plastic strain rate of $3 \cdot 10^6\ \mathrm{s}^{-1}$. Numerically, this was achieved by scaling the indenter stress field such that the collective motion of dislocations resulted in this plastic strain rate. All dislocation mechanisms implemented in the DDD code, e.g., formation of junctions and cross-slip, were taken into account.

We performed simulations for particles of size 24 nm, 48 nm, 72 nm, 96 nm and 120 nm, with the size referring to the distance between the substrate

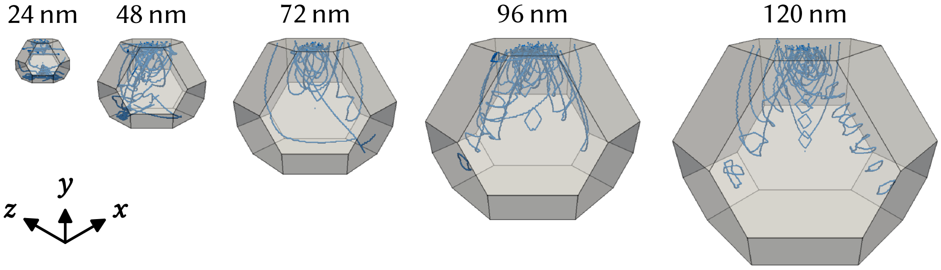


Figure 6.25.: Examples for dislocation microstructures obtained from indentation simulations for different specimen sizes.

interface and the surface being indented. For each size, we simulated 3856 realizations until a time of $1.5 \cdot 10^{-2}$ µs, corresponding to a plastic strain of 4.5 %. Exemplary dislocation microstructures are shown in Figure 6.25.

6.3.1.2. Extraction of dislocation microstructure features

We only consider the part of the dislocation microstructure contained within a spherical region of diameter 24 nm beneath the indenter, which is shown in Figure 6.24 as blue sphere. The FCC crystal structure exhibits mirror symmetry through the $\{1\,1\,0\}$ planes and three-fold rotation symmetry about the $\langle 1\,1\,1 \rangle$ directions, one of which is the indentation direction $[\bar{1}\,\bar{1}\,\bar{1}]$. This enables us to augment the number of observations for the machine learning models by a factor of six, leading to 23 136 observations per particle size.

We use spatial discretizations that are based on spherical polyhedra and show the same symmetries along the $[\bar{1}\,\bar{1}\,\bar{1}]$ direction to discretize the spherical region beneath the indenter. This means that for certain orientations, their symmetry aligns with the one of the crystal structure. These spatial discretizations are shown in Figure 6.26, with the $[\bar{1}\,\bar{1}\,\bar{1}]$ direction being parallel to the viewing direction, i.e., the spatial discretizations are viewed in indentation direction. The sphere-like spatial discretization contains one subdomain whose volume is equivalent to that of a cube with an edge length of 19.2 nm. The two tetrahedron-like spatial discretizations contain four subdomains, and they are distinguished depending on whether a subvolume is beneath the indenter contact point, or an edge where three subdomains meet. Each subdomain has a volume equal to that of a cube whose edge length is 12.1 nm. The icosahedron-like spatial discretization contains 20 subdomains of equal volume that is equivalent to that of a cube with an edge length of 7.1 nm. With the exception of the spherical discretization that is rotationally invariant, we also use variants of these spatial discretizations that are rotated

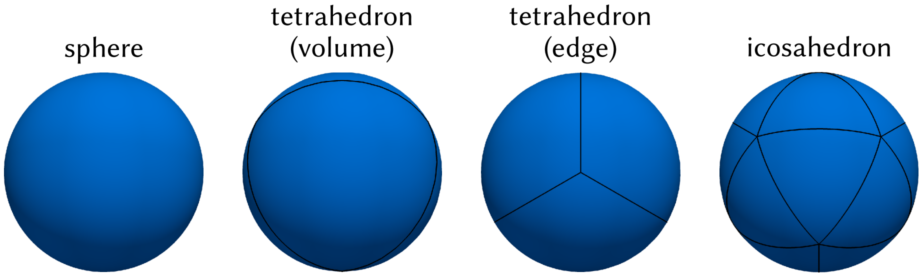


Figure 6.26.: Spatial discretization of the region beneath the indenter viewed along the $[\bar{1}\,\bar{1}\,\bar{1}]$ direction, i.e., the indentation direction.

about the $[\bar{1}\,\bar{1}\,\bar{1}]$ axis. We do this to study the effect of aligning the mesh with the crystal structure, as some rotations align with it while others do not.

We apply Jarek's subdivision algorithm once before fitting B-splines to the dislocation curves to account for the equal importance segment-based discretization of the microMegas DDD puts on all points along the segments. Subsequently, we carried out the numerical integration within the D2C method using a uniform evaluation point density of $2.5\,\mathrm{nm}^{-1}$ along the curves. Only independent components of the (partially) symmetric tensors were used to avoid duplicate features for the machine learning models.

6.3.1.3. Machine learning

We performed five shuffled stratified five-fold cross-validations for each combination of spatial discretization and dislocation microstructure features to evaluate the performance of the machine learning models. As for the relaxation experiment, we consider the GNB, SGD and HGBM machine learning algorithms as base for our machine learning models. To gain insight into how many observations are required to reach the desired accuracy, we computed learning curves. Similarly to the relaxation experiment, we also computed the permutation feature importance for each field in an attempt to understand how the machine learning models uses the available data to draw its conclusions.

6.3.2. Results

6.3.2.1. Average total dislocation densities

The average total dislocation densities for all spatial discretizations and specimen sizes are shown in figures 6.27 and 6.28. Each spatial discretization is shown from the indenter side and from the substrate side, in the left and

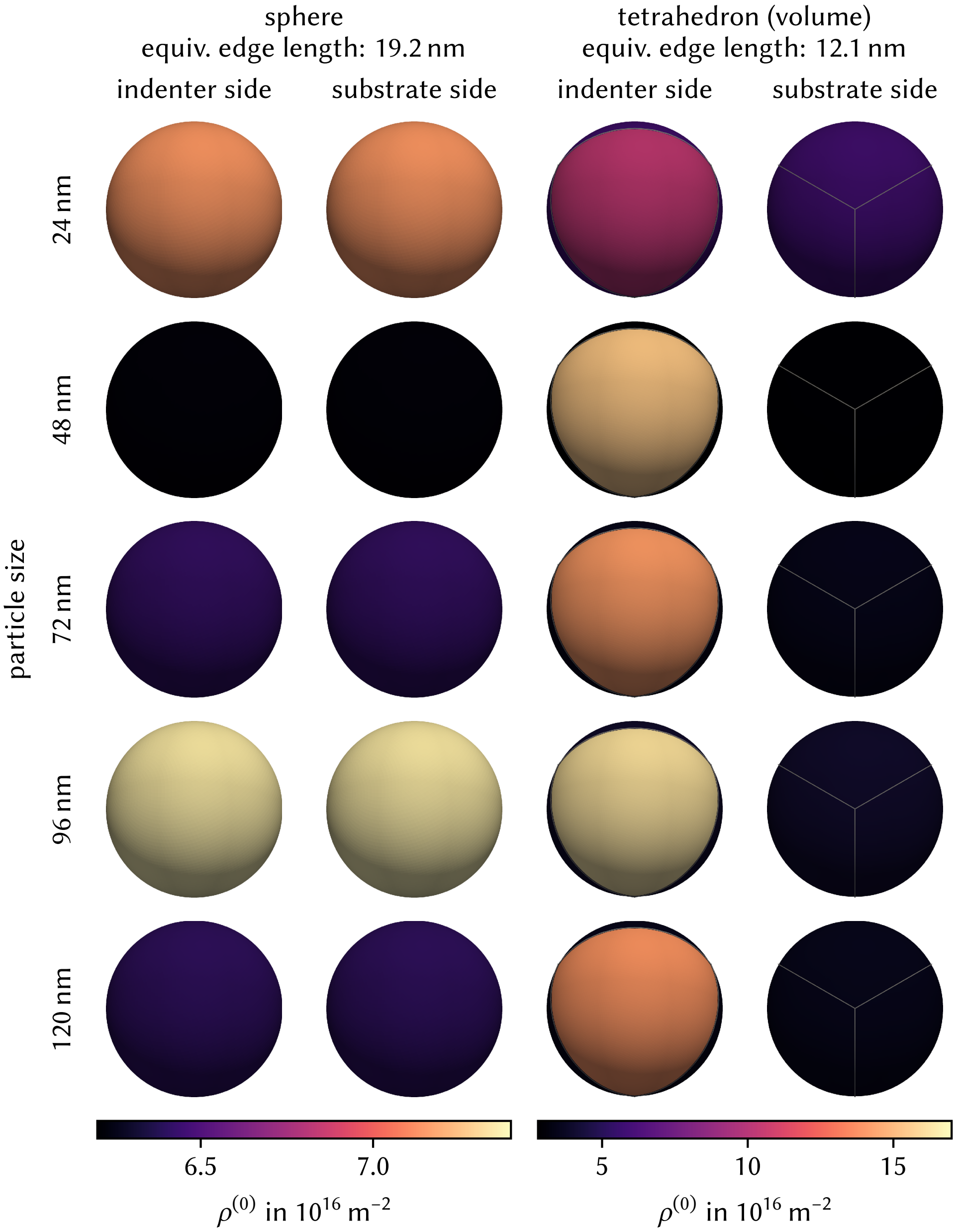


Figure 6.27.: Average total dislocation density of the sphere-like and tetrahedron-like (volume) spatial discretizations for different particle sizes. Both the view from the indenter side as well as the substrate side are shown.

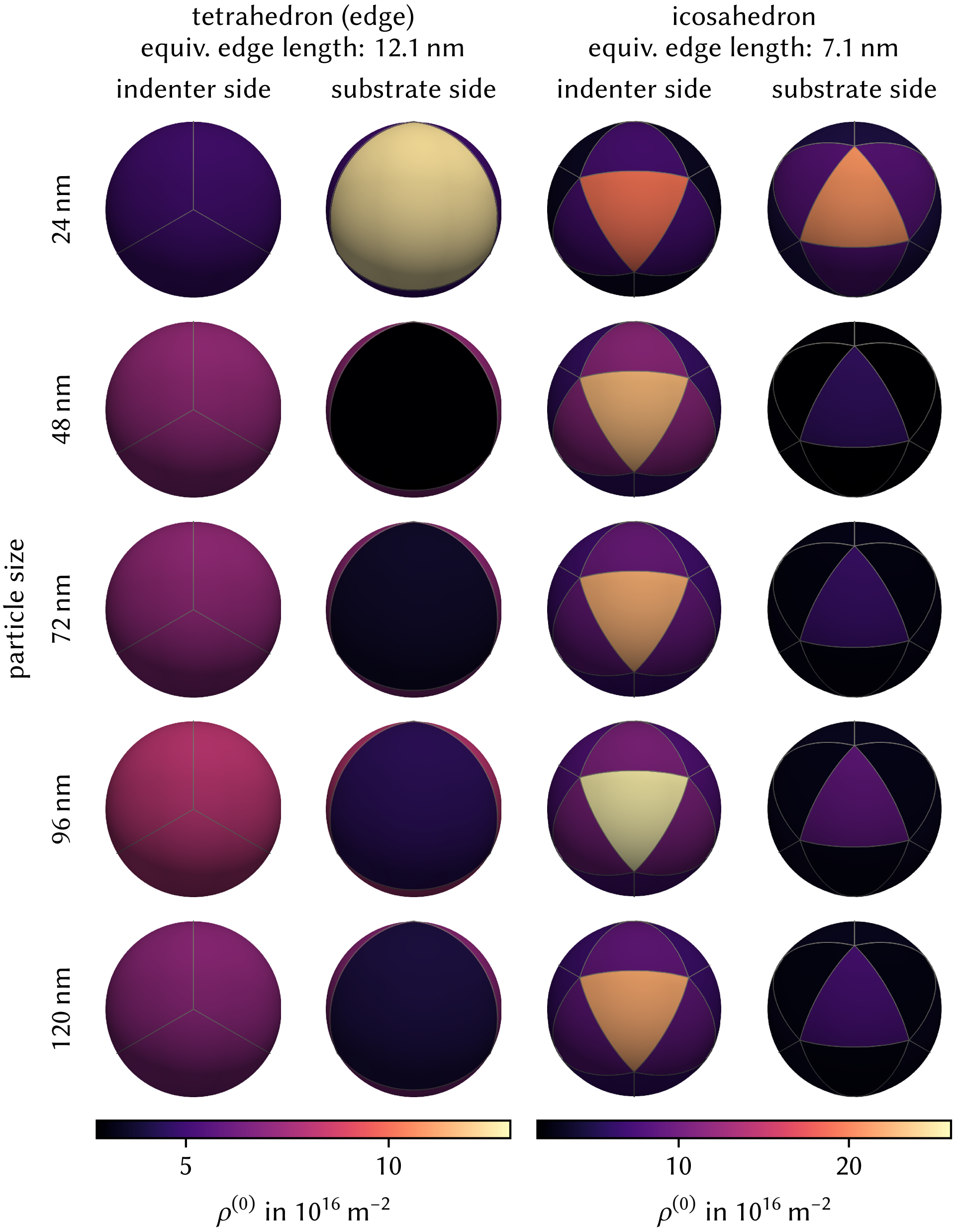


Figure 6.28.: Average total dislocation density of the tetrahedron-like (edge) and icosahedron-like spatial discretizations for different particle sizes. Both the view from the indenter side as well as the substrate side are shown.

Table 6.1.: Value range of the total dislocation density for spatial discretizations of the subvolume whose dislocation microstructure is used for classification.

spatial discretization	$\rho^{(0)}$ range
sphere-like	$6.2 \cdot 10^{16}\,\mathrm{m}^{-2}$ to $7.4 \cdot 10^{16}\,\mathrm{m}^{-2}$
tetrahedron-like (volume)	$2.8 \cdot 10^{16}\,\mathrm{m}^{-2}$ to $1.7 \cdot 10^{17}\,\mathrm{m}^{-2}$
tetrahedron-like (edge)	$2.8 \cdot 10^{16}\,\mathrm{m}^{-2}$ to $1.3 \cdot 10^{17}\,\mathrm{m}^{-2}$
icosahedron-like	$1.7 \cdot 10^{16}\,\mathrm{m}^{-2}$ to $2.6 \cdot 10^{17}\,\mathrm{m}^{-2}$

right column respectively. Rows represent the size of the nanoparticle whose average total dislocation density was computed. Colors denote the average total dislocation density within the subdomain of each spatial discretization. Samples taken from the 96 nm specimens exhibit the highest total dislocation density, followed in order by 24 nm, 72 nm, 120 nm and 48 nm. Generally, the subvolumes closer to the indenter exhibit higher average total dislocation densities than the subvolumes closer to the substrate. Samples taken from 24 nm specimens exhibit a large average total dislocation density on the substrate side compared to larger specimens, as shown in the substrate side columns of Figure 6.28. The range of the total dislocation density for each spatial discretization is summarized in Table 6.1.

6.3.2.2. Performance of the machine learning models

The accuracy scores for different field combinations and different spatial discretizations are shown in Figure 6.29 for GNB, in Figure 6.30 for SGD, and in Figure 6.31 for HGBM. We first cover observations specific to each algorithm. Subsequently, we compare the results across them.

Gaussian naïve Bayes

For the models based on the GNB algorithm, the single field performance ranked from worst to best is $q^{(0)}$, $\rho^{(0)}$, $\boldsymbol{\rho}^{(1)}$, $\boldsymbol{\rho}^{(2)}$, $\boldsymbol{\alpha}$, $\boldsymbol{\rho}^{(5)}$, $\boldsymbol{\rho}^{(3)}$, and $\boldsymbol{\rho}^{(4)}$. The inclusion of higher-order CDD fields increases the performance of the model. If the highest-order of CDD fields included is at least four, then not including all lower-order CDD fields is better. Adding the dislocation curvature density is beneficial when at least the 2nd-order CDD field is included as well. A higher spatial resolution leads to better model performance. For lower resolutions, substituting $\boldsymbol{\rho}^{(1)}$ by the Kröner–Nye tensor $\boldsymbol{\alpha}$ positively affects the performance of the models. More generally, field combinations that score best with the sphere-like spatial discretization take the Burgers vector into account. While still performing

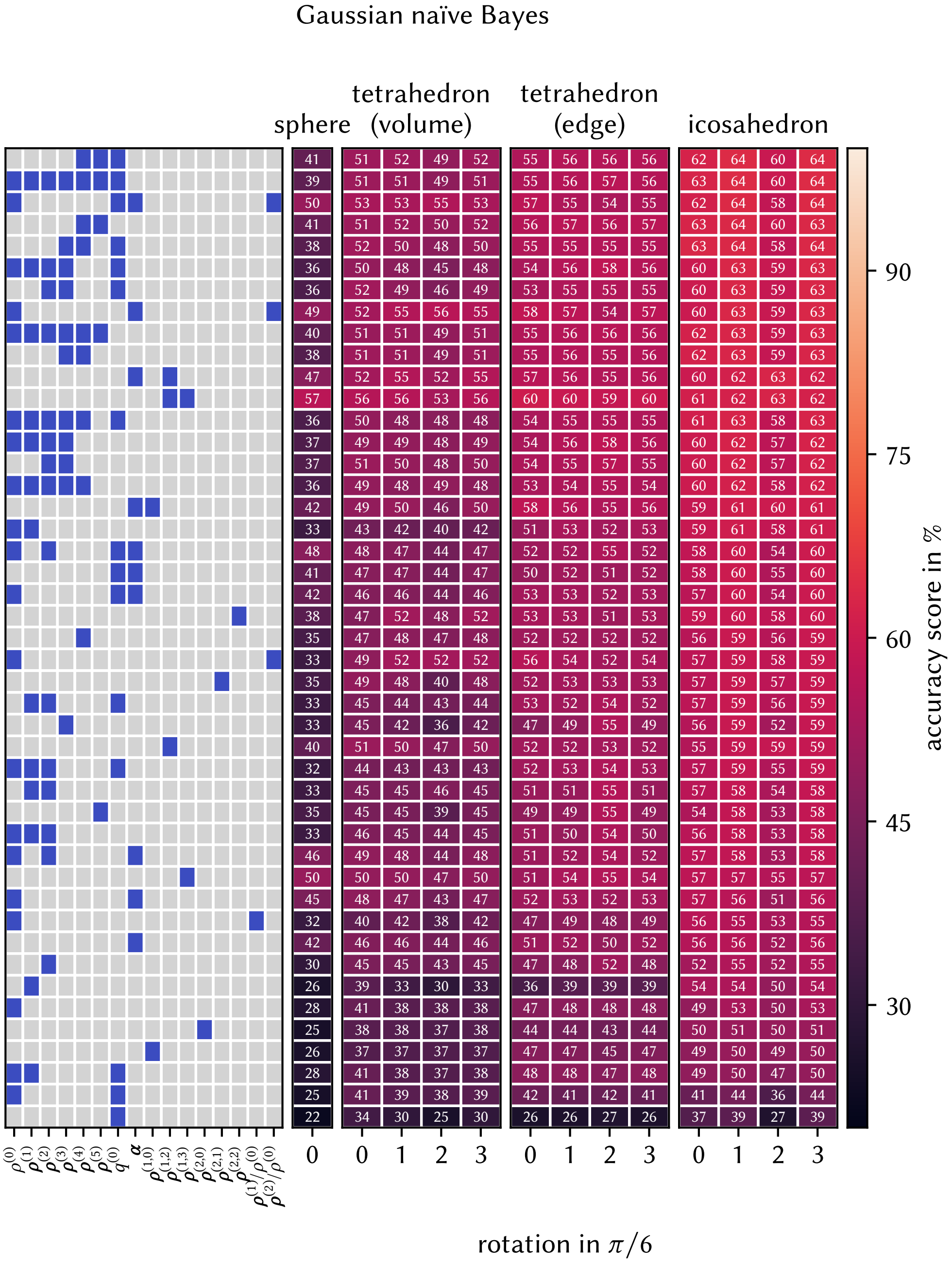


Figure 6.29.: Accuracy scores of the GNB-based machine learning models for different field combinations and different spatial discretizations. Rows are ordered by the highest accuracy score achieved by the respective field combination.

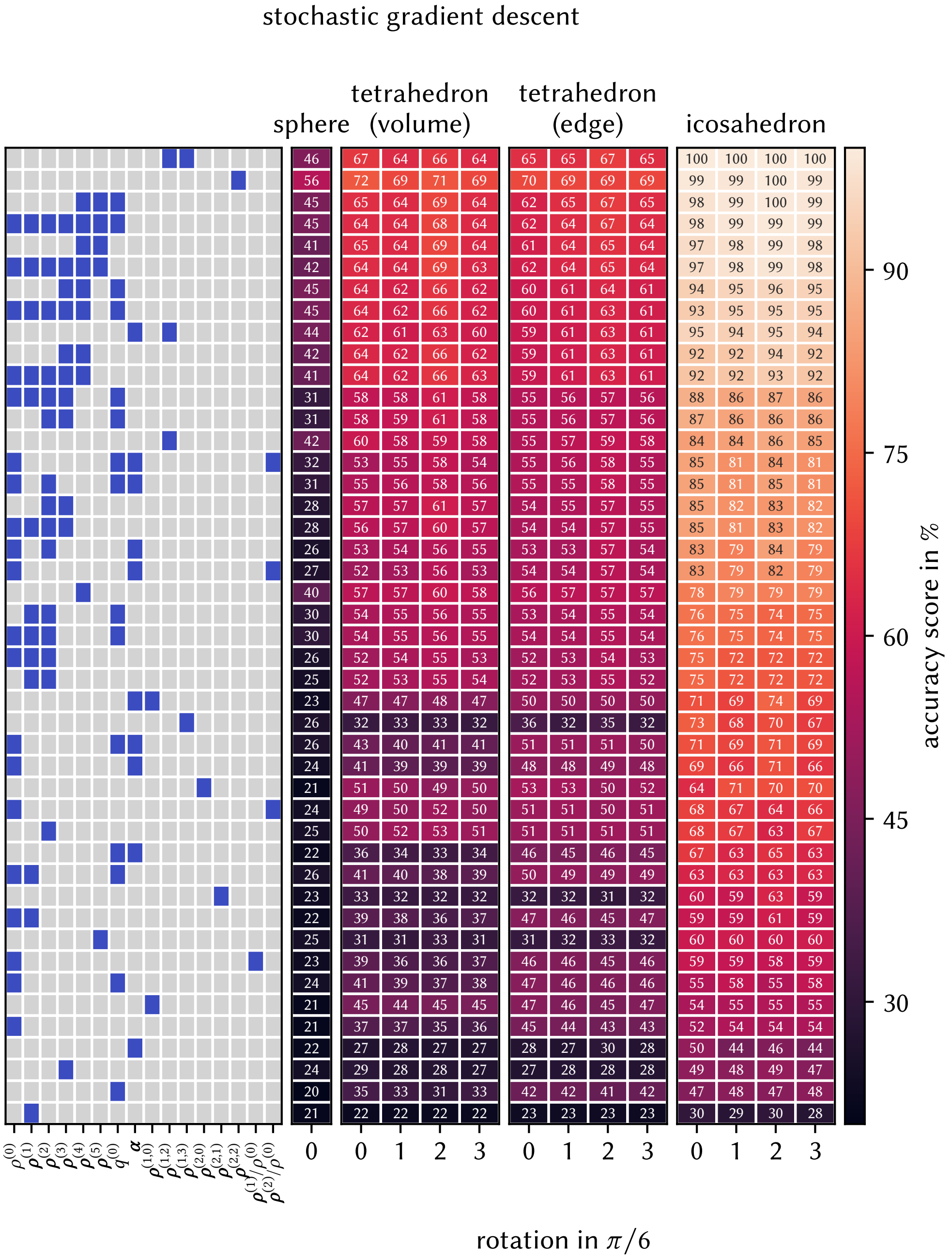


Figure 6.30.: Accuracy scores of the SGD-based machine learning models for different field combinations and different spatial discretizations. Rows are ordered by the highest accuracy score achieved by the respective field combination.

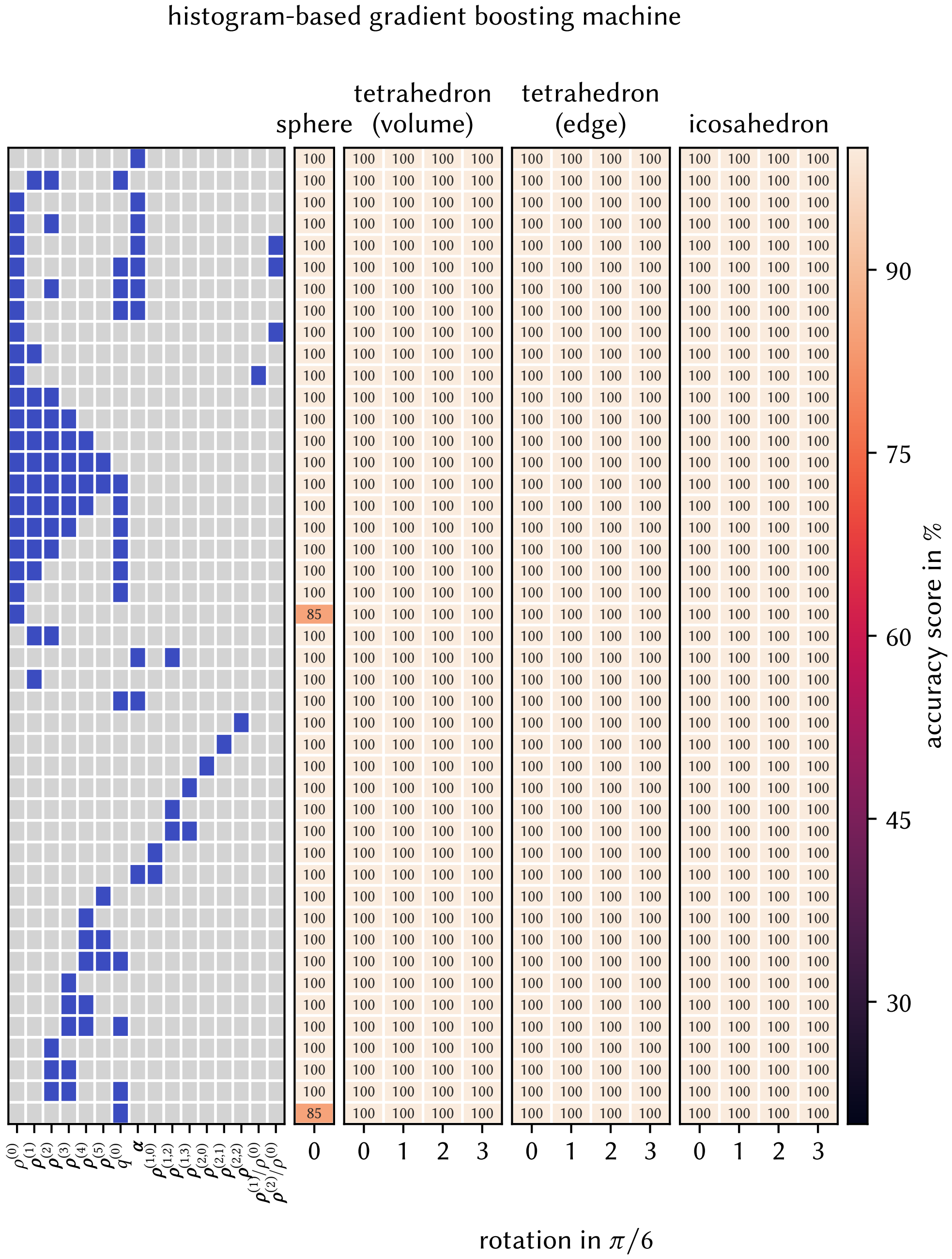


Figure 6.31.: Accuracy scores of the HGBM-based machine learning models for different field combinations and different spatial discretizations. Rows are ordered by the highest accuracy score achieved by the respective field combination.

well for the icosahedron-like spatial discretization, aforementioned trend is not observed for this higher resolution. For the best performing models using the icosahedron-like discretization, rotating the spatial discretization by $\pi/6$ and $\pi/2$ leads to better performance than the other two rotations.

Stochastic gradient descent
For the models based on the SGD algorithm, the single field performance ranked from worst to best is $\boldsymbol{\rho}^{(1)}$, $q^{(0)}$, $\boldsymbol{\rho}^{(3)}$, $\boldsymbol{\alpha}$, $\rho^{(0)}$, $\boldsymbol{\rho}^{(5)}$, $\boldsymbol{\rho}^{(2)}$, and $\boldsymbol{\rho}^{(4)}$. Using higher-order CDD fields improves the performance of the model. Adding the $\boldsymbol{q}^{(n)}$ has a positive impact on the performance, regardless of which other CDD fields are used. Leaving the lower-order fields out of the combination is better for the model quality if at least the 4^{th}-order CDD field is included in the fields. Substituting the Kröner–Nye tensor $\boldsymbol{\alpha}$ for $\boldsymbol{\rho}^{(1)}$ has a positive effect on the model quality. Higher resolutions of the spatial discretization are beneficial, achieving average accuracy scores of more than 0.99 for field combinations $\{\boldsymbol{\rho}^{(4)}, \boldsymbol{\rho}^{(5)}, q^{(0)}\}$, $\{\boldsymbol{\rho}^{(1,2)}, \boldsymbol{\rho}^{(1,3)}\}$, and $\{\boldsymbol{\rho}^{(2,2)}\}$. Rotating the icosahedron-like spatial discretization by $\pi/6$ and $\pi/2$ leads to better performance than the other two rotations for well performing models.

Histogram-based gradient boosting machine
Models based on the HGBM are able to classify the dislocation microstructures perfectly in almost all cases, the exception being $\{\rho^{(0)}\}$, and $\{q^{(0)}\}$ for the sphere-like spatial discretization.

Across algorithms
The highest accuracy scores for each algorithm are

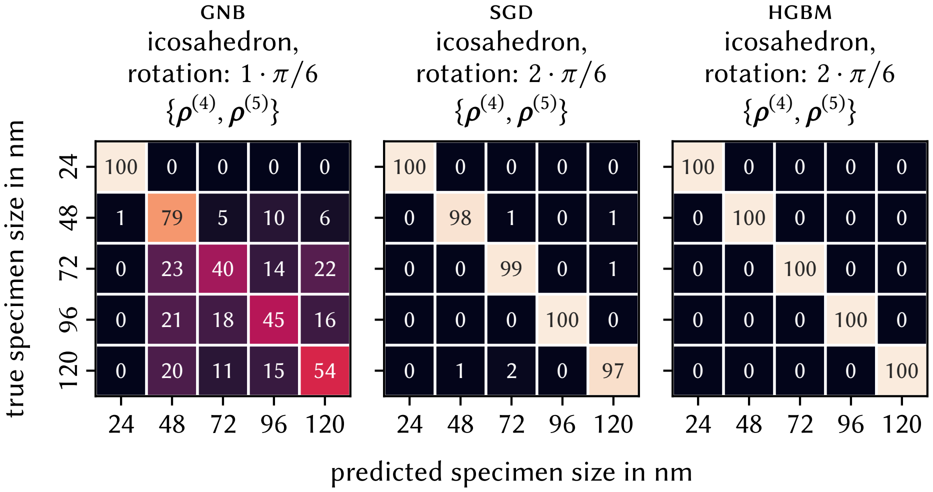
- 0.64 for the GNB with the fields $\{\boldsymbol{\rho}^{(4)}, \boldsymbol{\rho}^{(5)}, q^{(0)}\}$ using the icosahedron-like spatial discretization rotated by $\pi/6$, and
- 1.00 for the SGD with the fields $\{\boldsymbol{\rho}^{(1,2)}, \boldsymbol{\rho}^{(1,3)}\}$ using the icosahedron-like spatial discretization.

Due to the flawless accuracy in many different combinations, we can not point out a single best performing machine learning model based on the HGBM algorithm.

All employed algorithms benefit from a higher spatial discretization.

6.3.2.3. Confusion matrices

The confusion matrices of the best and worst performing subsequent CDD fields and spatial discretization for each algorithm are shown in Figure 6.32 and

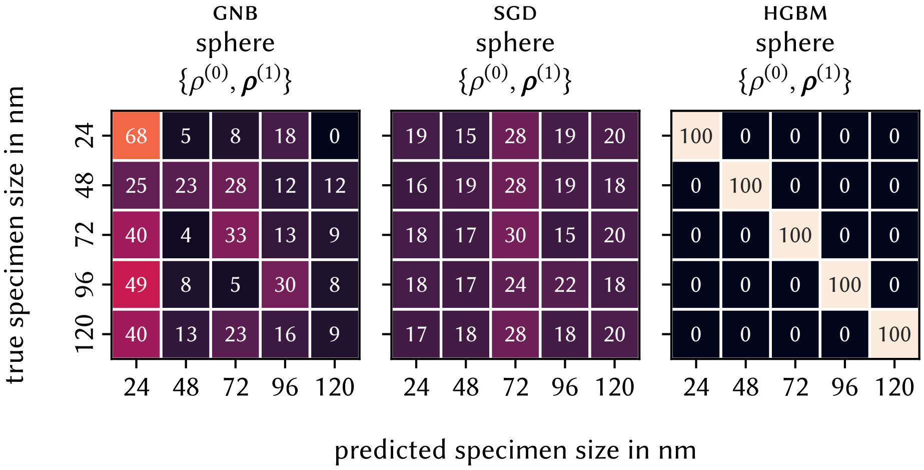


	24	48	72	96	120
24	100	0	0	0	0
48	1	79	5	10	6
72	0	23	40	14	22
96	0	21	18	45	16
120	0	20	11	15	54

	24	48	72	96	120
24	100	0	0	0	0
48	0	98	1	0	1
72	0	0	99	0	1
96	0	0	0	100	0
120	0	1	2	0	97

	24	48	72	96	120
24	100	0	0	0	0
48	0	100	0	0	0
72	0	0	100	0	0
96	0	0	0	100	0
120	0	0	0	0	100

Figure 6.32.: Confusion matrices of the best performing sequential CDD field and discretization combinations of all used machine learning algorithms.



	24	48	72	96	120
24	68	5	8	18	0
48	25	23	28	12	12
72	40	4	33	13	9
96	49	8	5	30	8
120	40	13	23	16	9

	24	48	72	96	120
24	19	15	28	19	20
48	16	19	28	19	18
72	18	17	30	15	20
96	18	17	24	22	18
120	17	18	28	18	20

	24	48	72	96	120
24	100	0	0	0	0
48	0	100	0	0	0
72	0	0	100	0	0
96	0	0	0	100	0
120	0	0	0	0	100

Figure 6.33.: Confusion matrices of the worst performing sequential CDD field and discretization combinations of all used machine learning algorithms.

Figure 6.33, respectively. As the HGBM exhibits perfect results in both cases, we do not outline its results further. Both the GNB-based and SGD-based models share the spatial discretization and field combination in these extreme cases. The only difference is the rotation of the icosahedron-like spatial discretization for the best performing field combination.

In case of the best performing subsequent fields, all algorithms are able to classify the dislocation microstructure of 24 nm specimens accurately. Additionally, the model based on the SGD algorithm furthermore classifies the ones of 96 nm specimens flawlessly.

For the worst performing subsequent fields, we can see preferences for the misclassification in both the GNB- and SGD-based models. The former tends to classify dislocation microstructures as coming from the 24 nm specimens and only few from the 120 nm one. The latter shows a preference for classifying dislocation microstructures to belong to the 72 nm specimens. In general, the misclassifications of the SGD-based models are more balanced than the ones of the GNB-based models.

The icosahedron-like spatial discretization generally leads to better performance of the machine learning models. Thus, we want to consider the difference between two field combinations that comprise two consecutive-order CDD fields but different order between them to study the impact of using higher-order CDD fields on *how* misclassification occurs. The resulting confusion matrices of the field combinations $\{\boldsymbol{\rho}^{(1)}, \boldsymbol{\rho}^{(2)}\}$, and $\{\boldsymbol{\rho}^{(3)}, \boldsymbol{\rho}^{(4)}\}$ are shown in Figure 6.34. All shown machine learning models are able to accurately classify the dislocation microstructure of 24 nm specimens. In all cases, the order of success in their accurate classification are 24 nm, 48 nm, 96 nm, 120 nm, and 72 nm. On the one hand, GNB-based models show a preference of misclassifying the remaining dislocation microstructures as belonging to the 48 nm specimens. On the other hand, GNB-based models show a preference of misclassifying dislocation microstructures of the 72 nm and 120 nm specimens more frequently than the ones of 48 nm and 96 nm.

Lastly, we want to study the confusion matrices when the whole sample is characterized by only the total dislocation density, which is shown in Figure 6.35. This combination performed poorly within all underlying algorithms.

In case of the GNB algorithm, we see a rather strong preference of classifying dislocation microstructures as belonging to the 24 nm sample. This effect is slightly weaker for the microstructures obtained from 48 nm specimens. There, classification is somewhat balanced between 24 nm, 48 nm and 72 nm.

For the SGD algorithm, the frequency of prediction varies little for different true specimen sizes. With about 0.29, most microstructures are classified as being obtained from 96 nm specimens. This is followed by 120 nm with about

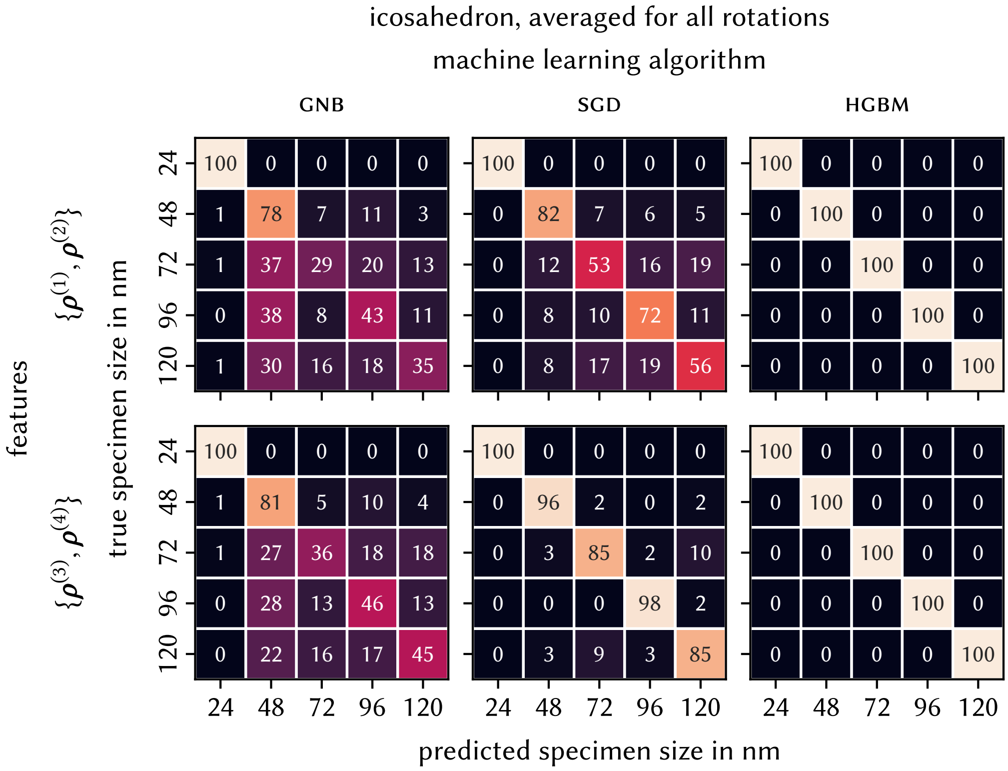


Figure 6.34.: Confusion matrices of the sequential CDD field combinations $\{\rho^{(1)}, \rho^{(2)}\}$, and $\{\rho^{(3)}, \rho^{(4)}\}$ for the icosahedron-like spatial discretization of all used machine learning algorithms. The values were averaged for all rotations.

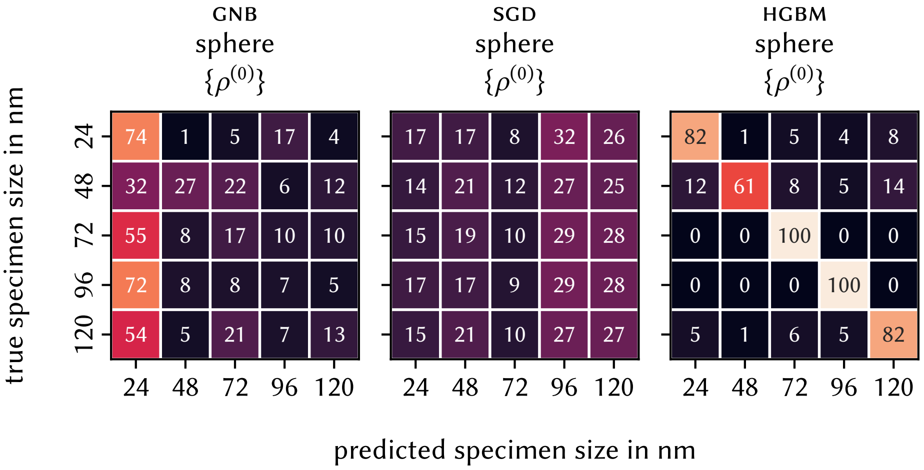


Figure 6.35.: Confusion matrices of using the total dislocation density $\rho^{(0)}$ as sole field and discretization combinations of all used machine learning algorithms.

0.27, 48 nm with about 0.19, 24 nm with about 0.16, and finally 72 nm with about 0.10.

The HGBM algorithm classifies dislocation microstructures from both 72 nm and 96 nm perfectly. Wrong classified microstructures exhibit a slight preference of being classified as belong to 24 nm, 72 nm and 120 nm, albeit not as strong as the preferences seen in the other two algorithms.

6.3.2.4. Feature importance

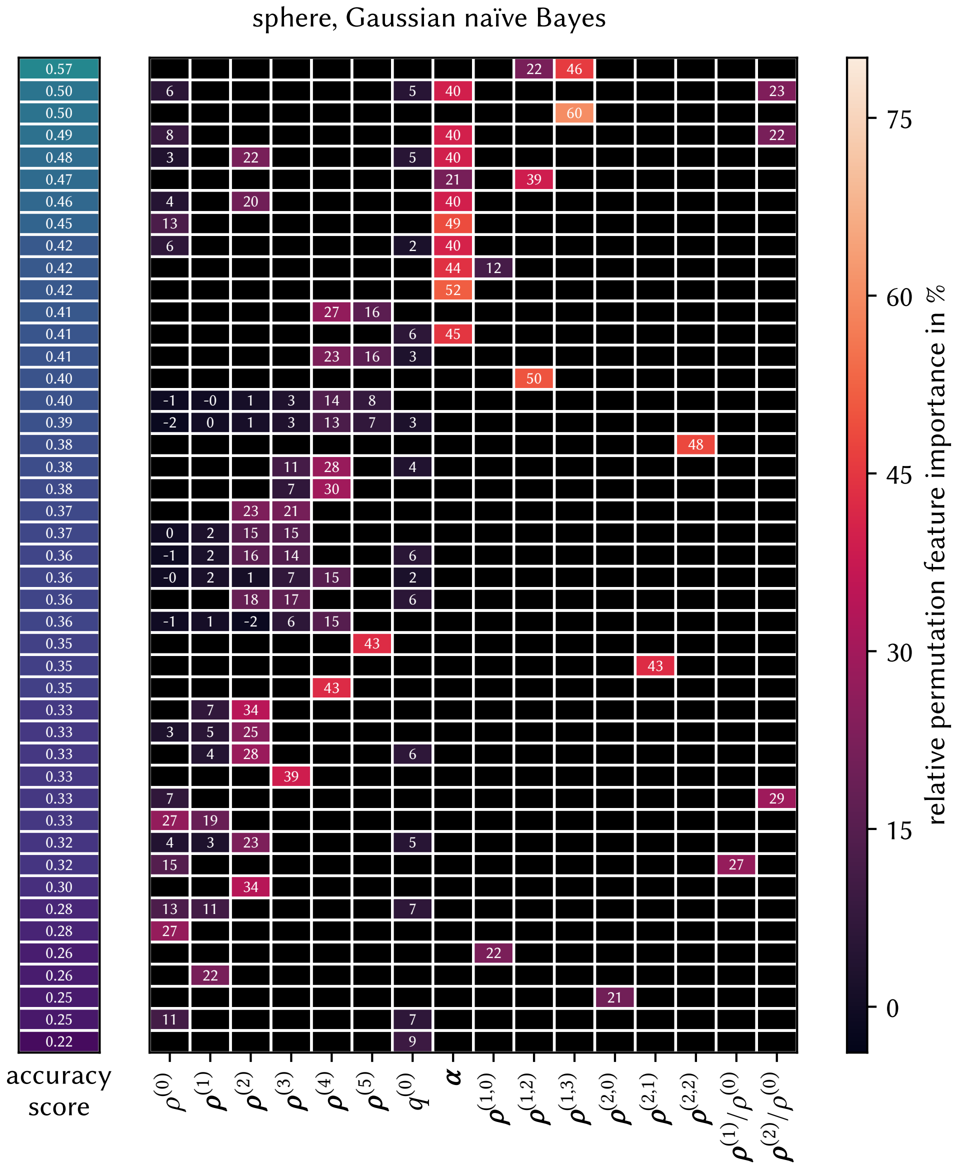
The relative permutation feature importances for different field combinations, spatial discretizations and models are shown in figures 6.36 to 6.41.

Gaussian naïve Bayes

When the sample is considered as a whole using the sphere-like spatial discretization, field combinations which take the Burgers vector into account by including at least one $\rho^{(m,n)}$ with $m = 1$ perform best. Within those field combinations, the fields containing the Burgers vector information are among the most important ones. For field combinations comprising only CDD fields, the highest even-order field contributes the most.

For the higher spatial resolution of the icosahedron-like spatial discretization, CDD fields perform on the same level as the fields taking the Burgers vector into account. Of the former, the most important are the highest even-ordered ones.

Irrespective of the spatial discretization, the dislocation curvature density



accuracy score	$\rho^{(0)}$	$\rho^{(1)}$	$\rho^{(2)}$	$\rho^{(3)}$	$\rho^{(4)}$	$\rho^{(5)}$	$q^{(0)}$	α	$\rho^{(1,0)}$	$\rho^{(1,2)}$	$\rho^{(1,3)}$	$\rho^{(2,0)}$	$\rho^{(2,1)}$	$\rho^{(2,2)}$	$\rho^{(1)}/\rho^{(0)}$	$\rho^{(2)}/\rho^{(0)}$
0.57										22	46					
0.50	6						5	40								23
0.50											60					
0.49	8							40								22
0.48	3		22				5	40								
0.47								21		39						
0.46	4		20					40								
0.45	13							49								
0.42	6						2	40								
0.42								44	12							
0.42								52								
0.41					27	16										
0.41							6	45								
0.41					23	16	3									
0.40										50						
0.40	-1	-0	1	3	14	8										
0.39	-2	0	1	3	13	7	3									
0.38														48		
0.38				11	28		4									
0.38				7	30											
0.37			23	21												
0.37	0	2	15	15												
0.36	-1	2	16	14			6									
0.36	-0	2	1	7	15		2									
0.36			18	17			6									
0.36	-1	1	-2	6	15											
0.35						43										
0.35													43			
0.35					43											
0.33		7	34													
0.33	3	5	25													
0.33		4	28				6									
0.33				39												
0.33	7															29
0.33	27	19														
0.32	4	3	23				5									
0.32	15														27	
0.30			34													
0.28	13	11					7									
0.28	27															
0.26									22							
0.26		22														
0.25												21				
0.25	11						7									
0.22							9									

Figure 6.36.: Relative permutation feature importance of each employed field for all field combinations for the sphere-like spatial discretization for the GNB-based machine learning models. The values are averages for all rotations of the spatial discretization. The field combinations are sorted by the achieved accuracy score.

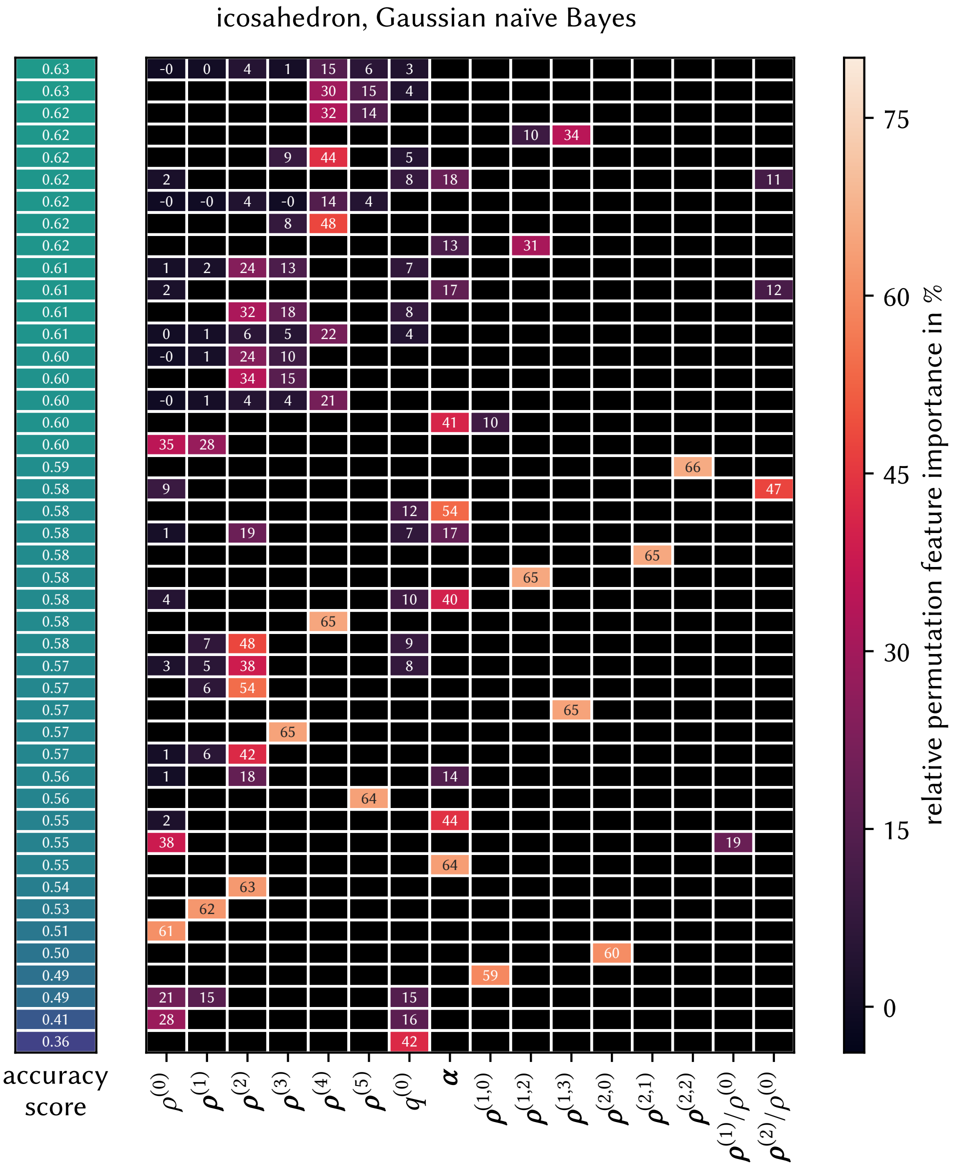
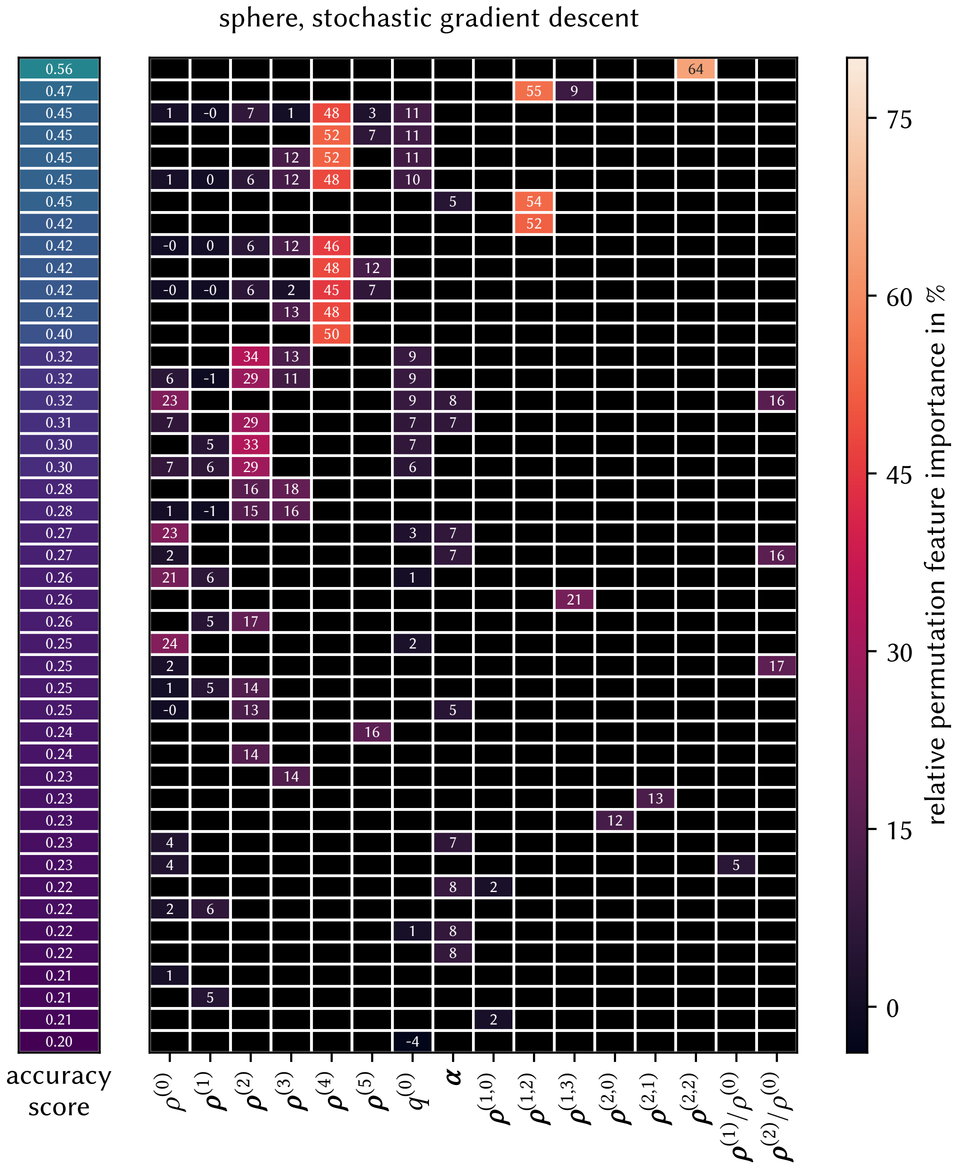

icosahedron, Gaussian naïve Bayes

Figure 6.37.: Relative permutation feature importance of each employed field for all field combinations for the icosahedron-like spatial discretization for the GNB-based machine learning models. The values are averages for all rotations of the spatial discretization. The field combinations are sorted by the achieved accuracy score.



accuracy score	$\rho^{(0)}$	$\rho^{(1)}$	$\rho^{(2)}$	$\rho^{(3)}$	$\rho^{(4)}$	$\rho^{(5)}$	$q^{(0)}$	α	$\rho^{(1,0)}$	$\rho^{(1,2)}$	$\rho^{(1,3)}$	$\rho^{(2,0)}$	$\rho^{(2,1)}$	$\rho^{(2,2)}$	$\rho^{(1)}/\rho^{(0)}$	$\rho^{(2)}/\rho^{(0)}$
0.56														64		
0.47										55	9					
0.45	1	-0	7	1	48	3	11									
0.45					52	7	11									
0.45				12	52		11									
0.45	1	0	6	12	48		10									
0.45								5		54						
0.42										52						
0.42	-0	0	6	12	46											
0.42					48	12										
0.42	-0	-0	6	2	45	7										
0.42				13	48											
0.40					50											
0.32			34	13			9									
0.32	6	-1	29	11			9									
0.32	23						9	8								16
0.31	7		29				7	7								
0.30		5	33				7									
0.30	7	6	29				6									
0.28			16	18												
0.28	1	-1	15	16												
0.27	23						3	7								
0.27	2							7								16
0.26	21	6					1									
0.26											21					
0.26		5	17													
0.25	24						2									
0.25	2															17
0.25	1	5	14													
0.25	-0		13					5								
0.24						16										
0.24			14													
0.23				14												
0.23													13			
0.23												12				
0.23	4							7								
0.23	4														5	
0.22								8	2							
0.22	2	6														
0.22							1	8								
0.22								8								
0.21	1															
0.21		5														
0.21									2							
0.20							-4									

Figure 6.38.: Relative permutation feature importance of each employed field for all field combinations for the sphere-like spatial discretization for the SGD-based machine learning models. The values are averages for all rotations of the spatial discretization. The field combinations are sorted by the achieved accuracy score.

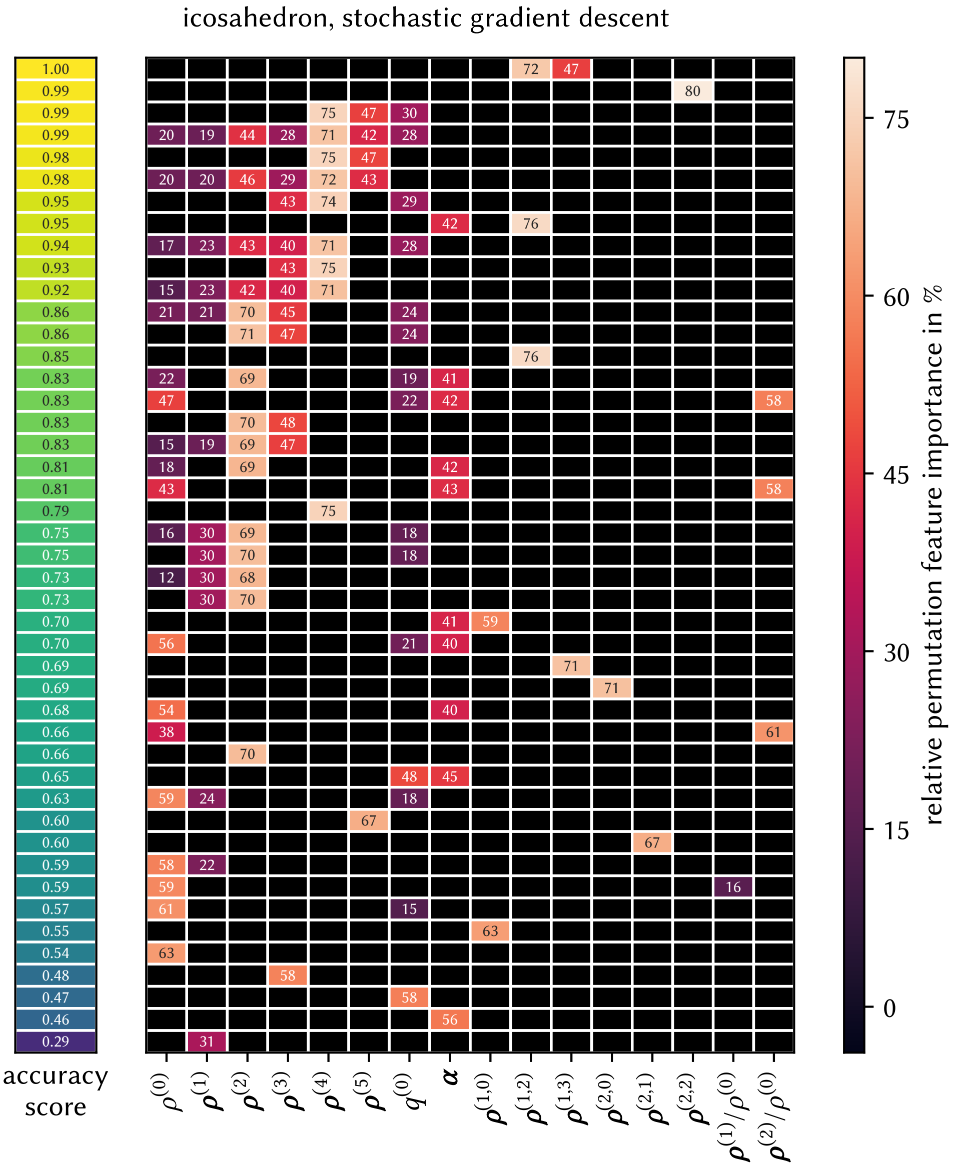


Figure 6.39.: Relative permutation feature importance of each employed field for all field combinations for the icosahedron-like spatial discretization for the SGD-based machine learning models. The values are averages for all rotations of the spatial discretization. The field combinations are sorted by the achieved accuracy score.

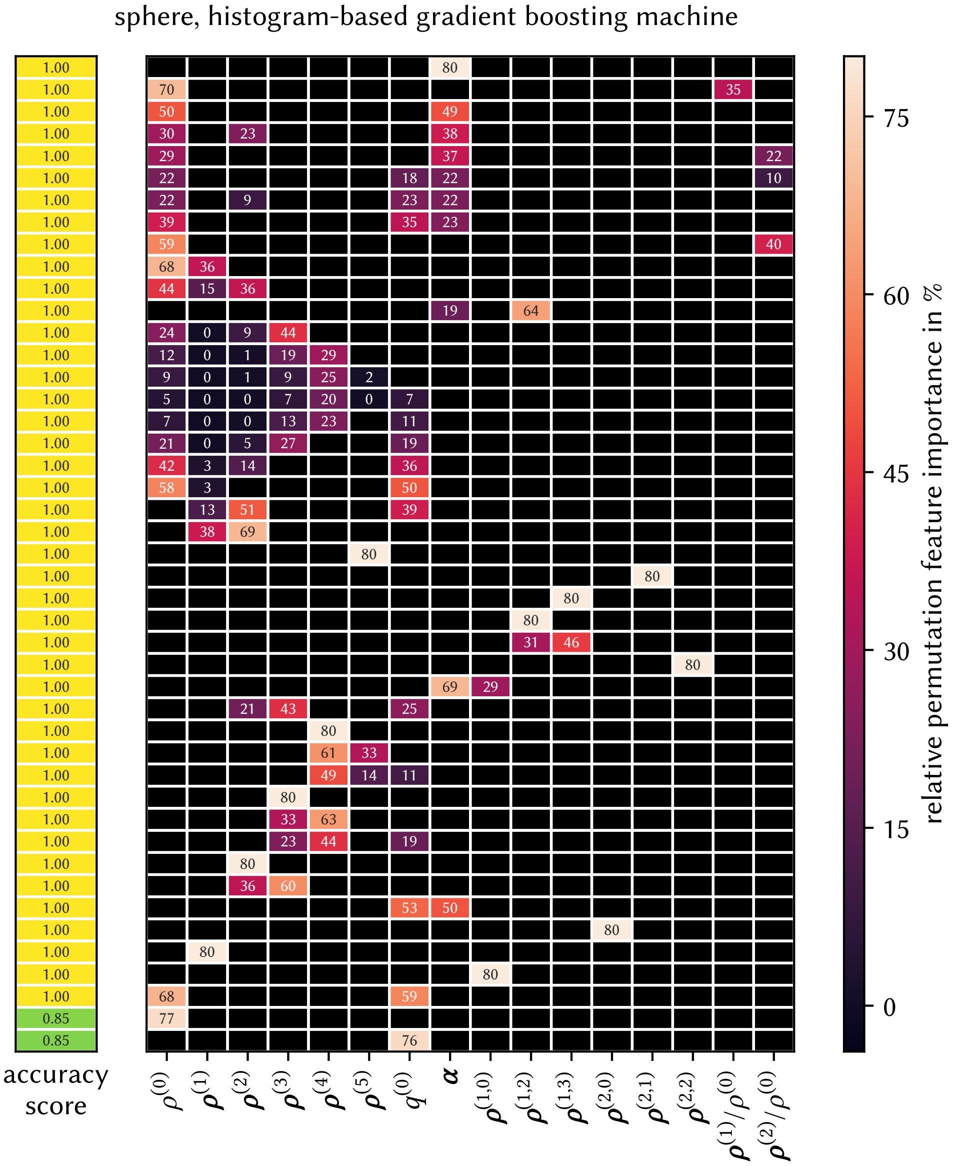


Figure 6.40.: Relative permutation feature importance of each employed field for all field combinations for the sphere-like spatial discretization for the HGBM-based machine learning models. The values are averages for all rotations of the spatial discretization. The field combinations are sorted by the achieved accuracy score.

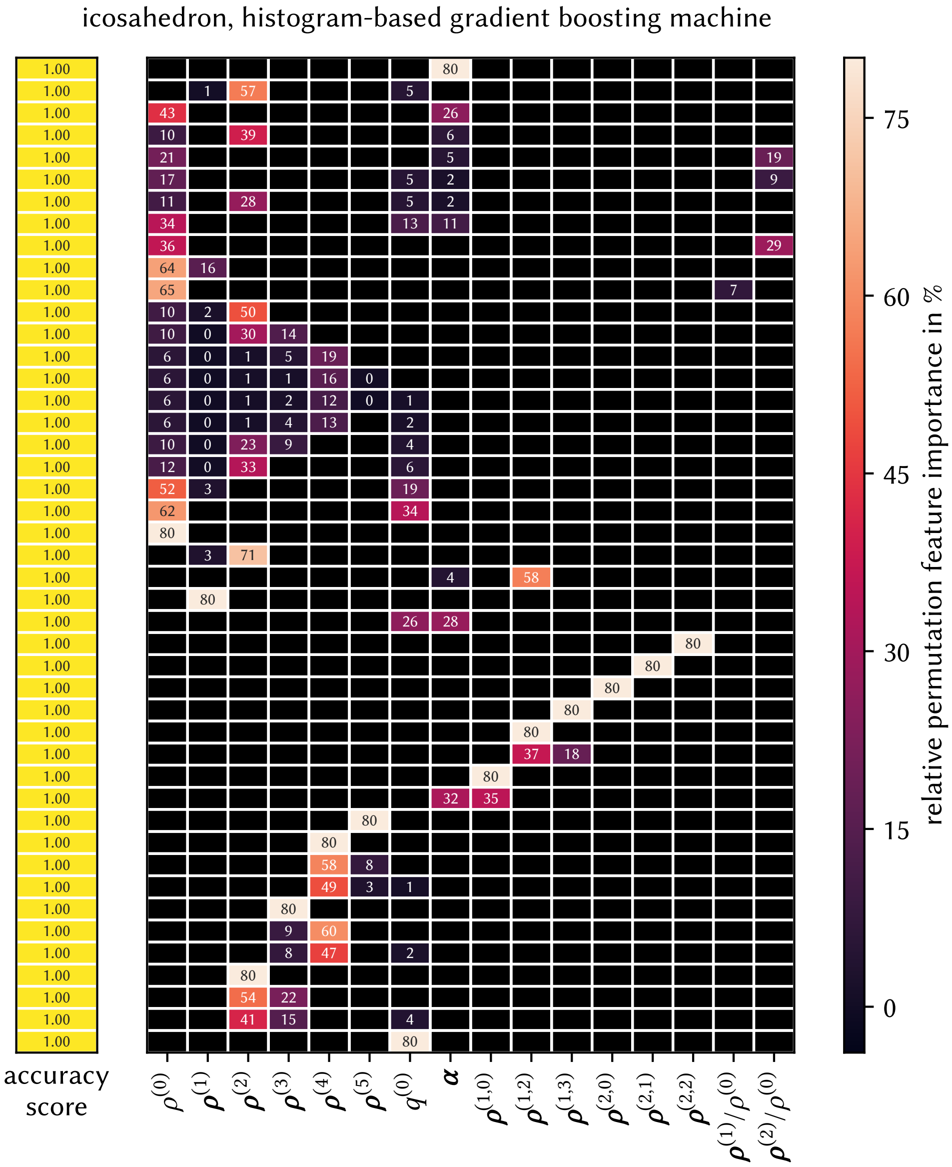


Figure 6.41.: Relative permutation feature importance of each employed field for all field combinations for the icosahedron-like spatial discretization for the HGBM-based machine learning models. The values are averages for all rotations of the spatial discretization. The field combinations are sorted by the achieved accuracy score.

contributes in relevant ways only for the field combinations including at most the 1st-order CDD density $\boldsymbol{\rho}^{(1)}$. Once at least $\boldsymbol{\rho}^{(2)}$ is included, the importance of the highest even-order CDD density is at least 2.5 times higher. Equally little important are the lower-order CDD fields that are included alongside the two highest-order fields.

Stochastic gradient descent
With regards to the CDD fields, we see a similar trend as for the GNB-based models. The highest even-order field is the most important by a wide margin compared to other fields. Compared to the GNB-based models, the dislocation curvature density, as well as odd-order and lower-order fields are more important for higher spatial resolution.

Histogram-based gradient boosting machine
Similarly to the previous algorithm, the HGBM puts most importance in the highest even-order CDD fields available to it. Odd-order ones are only relevant if they are both the highest-order CDD field and the order is smaller or equal three. The 5th-order CDD density hardly contributes to the success of models it is part of.

6.3.2.5. Learning curves

Learning curves for field combinations $\{\boldsymbol{\rho}^{(2)}, \boldsymbol{\rho}^{(3)}\}$ and $\{\boldsymbol{\rho}^{(4)}, \boldsymbol{\rho}^{(5)}\}$ for the icosahedron-like spatial discretization are shown in figures 6.42 and 6.43, respectively. For both field combinations, the GNB plateaus when more than 50 % of the available training observations are used, i.e., 46 272 observations. The machine learning models based on SGD improve up to using all observations within the training data set. After being exposed to more than about 5800 observations, HGBM reaches its accuracy score of 1.0.

6.3.3. Discussion

6.3.3.1. Accuracy of classifying dislocation microstructures of 24 nm specimen

The confusion matrices of icosahedron-like spatial discretizations show that the accurate classification of dislocation microstructures is more accurate for 24 nm specimen than for other specimen sizes. This likely stems from the boundary conditions used for the simulations. The surface in contact with the sapphire substrate, i.e., the one opposite of the indenter, is impenetrable for dislocations. Thus, dislocations form pile-up like structures at the bottom

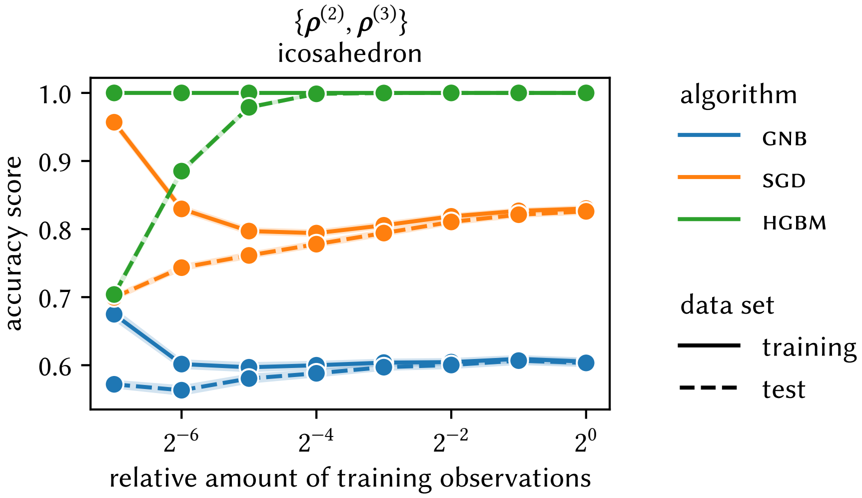


Figure 6.42.: Learning curve for the field combination $\{\rho^{(2)}, \rho^{(3)}\}$ and the icosahedron-like spatial discretization.

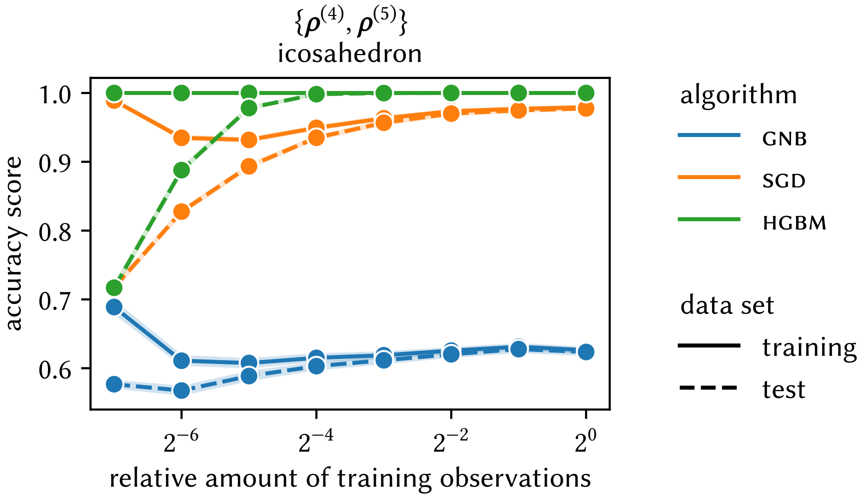


Figure 6.43.: Learning curve for the field combination $\{\rho^{(4)}, \rho^{(5)}\}$ and the icosahedron-like spatial discretization.

of the sample. Within 24 nm specimen, the subvolume whose dislocation microstructure is taken into account for the extraction of features covers this volume. The effect is nicely seen in Figure 6.28, on the substrate side view of the icosahedron-like spatial discretization for 24 nm. Compared to larger specimen, the subvolume closest to the substrate exhibits a distinctively large total dislocation density $\rho^{(0)}$. This is due to aforementioned dislocations which are unable to leave the specimen through the substrate boundary.

6.3.3.2. Including the deformation character of dislocations as features

In contrast to the previous relaxation experiment, we observe that the performance of the machine learning models depends more on the inclusion of both, odd-order CDD fields and fields that explicitly take the Burgers vector into account. This is because of the way we generated the dislocation microstructures. When we indent a nanoparticle, the occurring plastic deformation is carried by the dislocations that nucleate beneath the indenter and subsequently move into the particle. Thus, most of the dislocations would be considered GNDs.

The CDD fields that potentially take this aspect of dislocations into account are the odd-order dislocation density alignment tensors. In general, this is only valid when we use slip system specific odd-order CDD fields. We nevertheless observe that they are reasonably important for the machine learning models. To a certain extent, the implementation of slip systems within microMegas is the cause for this. We know that we could invert the Burgers vector of a dislocation without changing its properties when we also invert its sense. As the Burgers vector is not taken into account in CDD fields, this could lead to contributions that cancel each other out even though their deformation character might add up, simply due to their Burgers vectors being inverted. Within microMegas however, Burgers vectors are uniquely oriented. This means that dislocations of a slip systems with the same Burgers vector direction share the exact Burgers vector. Thus, the sense alone is the deciding factor for the configuration of the dislocations within these slip systems. This partially alleviates the necessity of splitting odd-order CDD fields into the present slip systems and explains why they are still able to contribute to the performance of the machine learning models.

Fields fully taking the Burgers vector into account are the Kröner–Nye tensor and the dislocation feature density tensors $\boldsymbol{\rho}^{(1,n)}$. The machine learning models based upon these fields exhibit performances at least comparable to that of higher-order CDD fields, e.g., the combination $\{\boldsymbol{\alpha} = \boldsymbol{\rho}^{(1,1)}, \boldsymbol{\rho}^{(1,2)}\}$ performs at least as good $\{\boldsymbol{\rho}^{(3)}, \boldsymbol{\rho}^{(4)}\}$. As they consider the Burgers vector, they fully

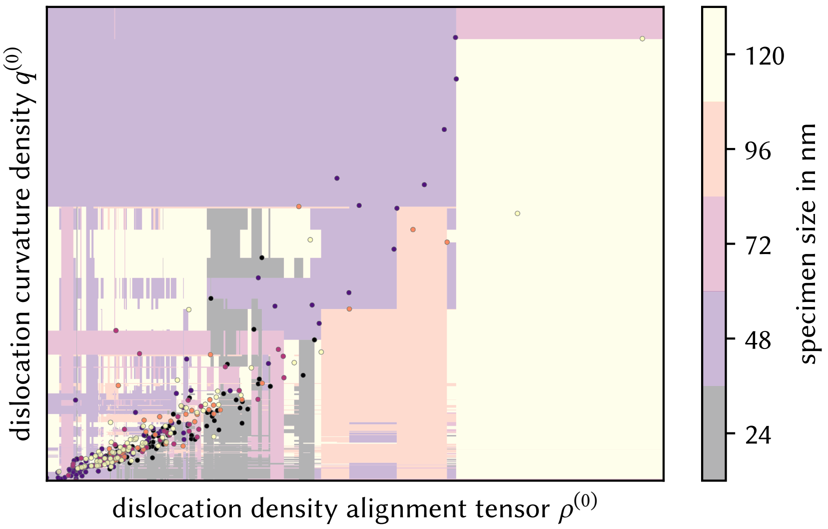


Figure 6.44.: Decision boundary of the machine learning model based on the HGBM utilizing the field combination $\{\rho^{(0)}, q^{(0)}\}$. Markers denote both training and test data used to train and test this machine learning model. The axis values are normalized such that the sample mean and sample standard deviation on the training data are zero and one, respectively.

take the deformation character of the GNDs into account. And due to most dislocations in this indentation experiment contributing to the deformation of the nanoparticle, machine learning models that have this information available ultimately perform better.

6.3.3.3. Performance of the HGBM algorithm

The machine learning models based on the HGBM algorithm perform flawlessly in almost all combinations of spatial discretization and fields. This is not something we would expect, especially after comparing these results with the ones obtained from the relaxation experiment. So how is it possible that it performs equally well on the test data? To get a better idea what the machine learning models might have learned, we show the decision boundary for the sphere-like spatial discretization and the field combination $\{\rho^{(0)}, q^{(0)}\}$ in Figure 6.44. This particular combination is able to reach an accuracy score of 1.00 with exactly 2 features, allowing us to visualize it easily. Besides the decision boundary, we also show the single observations used to both train and test the machine learning model.

The first thing we see is that the decision boundary is very fragmented, especially for combinations of low $\rho^{(0)}$ and $q^{(0)}$. As this combination also exhibits the highest concentration of observations, we may conclude that the machine learning model adapts well to the information provided by its training data.

Still, this only explains why it performs so well on the training data. If it was overfitting, then the test performance would be worse. So how does it perform equally as well on the test data? The reason is the spatial discretization used by the DDD code we used to evolve these dislocation microstructures. In this code, microMegas, dislocations are represented using line segments of fixed orientation and length. That way, a lattice of possible configurations for dislocation lines is formed. It also means that the density values we compute are discrete and can not take on arbitrary values. When we extract the features from the simulations, we do so in a spherical sample beneath the indenter. The position of this spherical sample is different for each specimen size. Thus, every spherical sample contains a particular cut of the lattice that dislocations can be on which is specific to each specimen size. Combining this information with the fact that the computed density values are discrete we come to the conclusion that the values we extract as feature result in disjoint sets of values for each specimen size. The learning curves support this claim. Regardless of how few observations we include in the training data, the accuracy score is perfect. Furthermore, the points denoting observations within Figure 6.44 are plotted with very low alpha value. This can be seen for some points; the edges of the points beneath them are visible. Most of them, however, seem to be opaque. This is due to many semi-transparent markers overlapping.

When the number of observations in the training data set is sufficiently large, then the machine learning model is exposed to most of the possible configurations for each specimen size. This means that also most observations of the test data set are not “new”, but have been part of the training data already. Note, that this does not mean that the dislocation microstructure for observations having the same features is exactly the same. It merely means that the exact configuration that makes them differentiable is lost in the conversion to continuous fields in the spatial discretizations we utilized. This aligns with the drop in the accuracy score we observe in the learning curves for small training data sets. While the training data set is still classified perfectly, not all possible feature values for each specimen size are encountered during training and therefore the test data set exhibits misclassifications.

The conclusion is that the machine learning models based on the HGBM algorithm do not learn underlying relationships between dislocation microstructure features. Instead, they learn that each observation obtained from a specific specimen size exhibits particular feature values that uniquely identify the specimen size because of the dislocation representation.

6.4. Discussion

6.4.1. Continuum fields as machine learning features

We studied two different situations in which dislocation microstructures with different characteristics form. On the one hand, we allowed a random dislocation microstructure to evolve without the application of an external load. Most dislocations in this case would be considered SSDs. On the other hand, we evolved a dislocation microstructure in an indentation experiment. In this case, most dislocations would be considered GNDs. Studying these two extreme cases gives us an idea about whether continuum dislocation density fields are sufficient as features, and how different field combinations affect the performance.

For the former case, the baseline accuracy score of 0.33 is substantially surpassed by the machine learning models, with many combinations of fields and machine learning algorithms reaching at least 0.80. In the latter case, the performance depended more on the combination of fields and spatial discretizations. Still, the baseline accuracy score of 0.20 is improved upon by many combinations reaching more than 0.60, for specific combinations also surpassing 0.80. For classification, at least, continuum fields of dislocations are viable features that lead to good performance.

Are they also viable features for regression problems? Based on the results obtained for classification so far, we argue that it strongly depends on the quality of the machine learning model. From the confusion matrices obtained in the indentation experiment, we observe that well performing models exhibit a slight preference of classifying the specimen size as adjacent to the actual specimen size.

The average effect of specific field combination modifications is summarized in Table 6.2. We computed the change in the accuracy score for each experiment and machine learning algorithm when different aspects of the field combinations are changed. This allows us to provide some general pointers for different experiments, as a thorough study of all possible field- and algorithm-combinations is not always feasible.

We observe that an increase in the order of the dislocation density alignment tensors fields always leads to an improvement in the performance of the machine learning model. As the order increases, so does the information contained in the resulting feature. And in most cases this allows the machine learning model to better learn the underlying relations of the dislocation microstructure with respect to the output we want to predict. Machine learning models based on the SGD algorithm are especially sensitive to an increase in

Table 6.2.: Average change of accuracy score for changes in the features of a machine learning model. The values are given in percentage points.

	relaxation			indentation		
	GNB	SGD	HGBM	GNB	SGD	HGBM
using higher-order $\boldsymbol{\rho}^{(n)}$	0.5	1.2	1.0	2.1	6.6	0.0
including lower-order $\boldsymbol{\rho}^{(n)}$	0.9	−0.1	0.0	−0.5	−0.1	0.0
replacing $\boldsymbol{\rho}^{(n)}$ by $\boldsymbol{\rho}^{(n)}/\rho^{(0)}$	0.1	3.4	0.3	2.1	−0.4	0.0
including $\boldsymbol{q}^{(n)}$	−9.6	3.4	0.3	−0.3	2.6	0.0
replacing $\boldsymbol{\rho}^{(n)}$ by $\boldsymbol{\rho}^{(1,n)}$	−3.5	−35.6	−4.7	4.5	8.7	0.1

the order. Our general recommendation would therefore be to use the highest-order CDD fields affordable within the available time and computational power constraints.

From equation (2.33) we know that all information of dislocation density alignment tensors up to n$^{\text{th}}$-order is contained within $\boldsymbol{\rho}^{(n-1)}$ and $\boldsymbol{\rho}^{(n)}$. Including the lower-order dislocation density alignment tensors in the field combination as well could have made sense as we do not explicitly inform the machine learning models about the relationship given in equation (2.33). However, from our experiments we observe that this does not positively affect the performance of the machine learning models except for the ones based on the GNB algorithm within the relaxation setting. It seems as if the other machine learning models are flexible enough to extract the information contained in the highest-order fields. We therefore recommend only taking the two highest-order CDD fields into account to save computational power and time.

6.4.2. Influence of the spatial discretization

We observe that the spatial resolution as well as the alignment with respect to crystallographic axes impact the performance of machine learning models. Coarser spatial resolution means that we average over more parts of the dislocation microstructure and put less emphasis on where exactly these parts are within the microstructure. For the relaxation experiment, we varied the edge length of the cubic subvolumes between 5 nm to 30 nm, and for the indentation experiment, we considered subvolumes with equivalent edge lengths of a cubic subvolume between 7.1 nm to 24 nm. The misorientation of spatial discretizations with respect to the crystallographic axes may change the likelihood with which we average over microstructure features that strongly depend on the crystal lattice, e.g., parts of dislocations that are on separate slip systems due to cross-slip.

GNB-based machine learning models show mixed results regarding the spatial discretization. In their case, the distribution of the features is crucial and if the spatial discretization enables separation of the feature distributions, then GNB performs well. This could mean that machine learning models built upon this algorithm show a large dependency on the spatial discretization for different dislocation microstructures.

In contrast, machine learning models based on the SGD machine learning algorithm benefit from finer spatial discretizations, regardless of the experiment. At least within the range of studied discretizations, we therefore recommend to use higher resolutions for the spatial discretization in order to increase the performance of SGD-based machine learning models.

Giving a more general recommendation for the HGBM is not possible due to the machine learning models learning an implementation detail of the DDD code in the indentation experiment instead of underlying relationships of the dislocation microstructure. From the relaxation experiment, however, we observe that the machine learning models are able to work well with low spatial resolution and would therefore start with a coarser discretization.

Something we have not explored in this work is combining different fields obtained from different spatial discretizations. Using higher-order fields on a coarse spatial discretization and combining it with lower-order fields on a fine spatial discretization might improve the machine learning model. The higher-order fields would then capture the finer details of the dislocation orientations within larger volumes, while the lower-order fields take the position of dislocations irrespective of their exact orientation within that larger volume into account. This way, small local fluctuations in orientation within similar overall structures over larger length scales would not lead to larger differences in their features.

6.4.3. Number of observations required for sufficient training

To ensure that we are using the potential of the investigated machine learning models, a minimum number of observations is required. From the learning curves determined for relatively feature efficient continuum dislocation density field and spatial discretization combinations, we may conclude that about 50 000 observations are required. This number is large, but considering the vast amount of possible dislocation configurations that are possible within even a small volume, it seems reasonable. A valid approach for saving computational resources and gaining the most from every simulation and experiment, we suggest augmenting the available data in every way possible, e.g., as shown in this work by exploiting symmetries of the crystal structure and the geometry of the specimen.

6.4.4. Considerations for a dislocation microstructure database

Machine learning models using features extracted via the D2C method can aid in gaining knowledge when a database containing dislocation microstructures and the accompanying metadata has been set up. Instead of identifying the size of a particle a specific dislocation microstructure belongs to, more general concepts might be studied.

One example could be the comparison of dislocation microstructures obtained from experiments and simulations. If we trained a machine learning model with the goal to correctly classify whether a dislocation microstructure originates from an experiment or from a simulation, then a performance close to the baseline would mean that the dislocation microstructures are indistinguishable, which essentially validates the correctness of the simulation code. This would be more of a one-off machine learning model that is used once to verify the simulation code.

Another example could be the using a machine learning model to approximate the elastic energy contribution in a subvolume to the presence of dislocations. In this case, the machine learning model could be continuously improved when new data is added to the database.

However, these applications may well require far more and more diverse data than the one we used in this work. And then the time it takes to train a machine learning model becomes important. The minimum time it took to train each machine learning models used in this work over the number of features is shown in Figure 6.45. For the same number of features the machine learning models based on the SGD machine learning algorithm require more than one order of magnitude more training time than ones based on the GNB machine learning algorithm. When using such machine learning models on larger scale data, we have to balance the feasibility of training a machine learning models within the available time window when considering one-off models. For a more general purpose machine learning model like the one in the second example outlined before, training times do not matter that much as long as the machine learning algorithm supports incremental learning. A

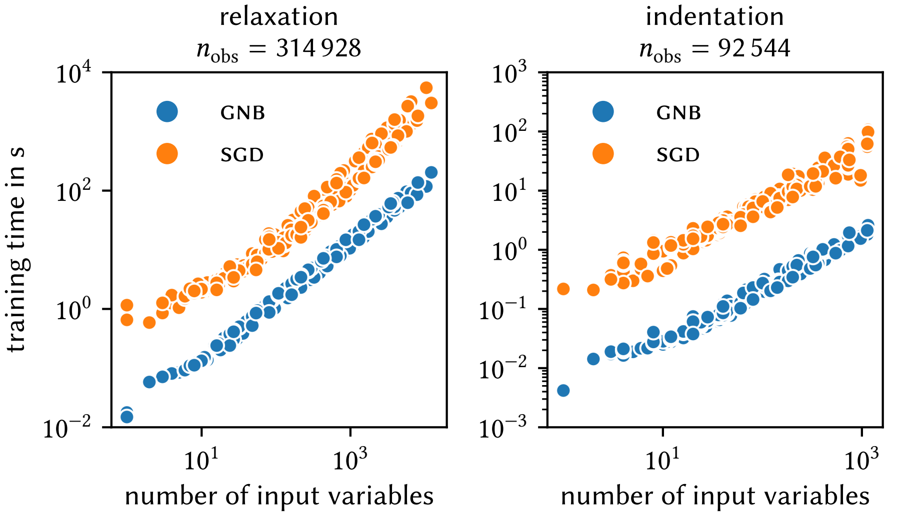


Figure 6.45.: Minimum training time over the number of featuress for all machine learning models used in this work.

machine learning model could be persisted on the server and be loaded and further trained whenever new data is entered in the database. This way, the quality of the machine learning model increases with the amount of data collected. In that case, we recommend increasing the amount of features to ensure the highest possible accuracy that continuously improves over time.

6.5. Summary and conclusion

We have generated dislocation microstructures within nanoparticles of different sizes in two different numerical experiments via DDD simulations: on the one hand, relaxation of random initial dislocation microstructures that resulted in microstructures comprising mostly SSDs, on the other hand, indentation with dynamic nucleation of dislocations that resulted in microstructures consisting mainly of GNDs. We used the D2C method to extract continuum dislocation density fields. These quantities were then used as input for machine learning models with the aim to classify the particle size a dislocation microstructure belongs to based on its features alone. The primary parameters studied were the combination of continuum dislocation density fields, the spatial discretization of the subvolume containing the dislocation microstructure, and the underlying machine learning algorithm. Their affect on the machine learning models performance was investigated using both accuracy score and confusion matrices. We have used the permutation feature importance to identify the fields and locations of features contributing the most to the performance of the machine learning models. Finally, by generating learning curves we were able to gain insight on how many observations are required to exploit capabilities of the machine learning models.

We found that continuum dislocation density fields are well suited as features within the context of machine learning, and that higher-order fields generally lead to better results. Whether the fields should include information about the Burgers vector or not depends on whether the dislocation microstructure comprises mostly SSDs or GNDs. Depending on the underlying machine learning algorithm, spatial discretizations may have to be chosen differently if optimal results are desired. Knowing the “inner workings” of the machine learning algorithms and the DDD codes was shown to be important if generalizable machine learning models should be developed. About 50 000 observations should be supplied for training the machine learning models to begin using them to their full potential.

7. Conclusion and outlook

First, we demonstrated how the combination of in situ experiments on the nanometer length scale with eigenstrain-based FEM analyses allowed us to reconstruct the full three-dimensional configuration of the emerging dislocation microstructure. Furthermore, we were able to fully characterize the slip systems of every dislocation.

Subsequently, we studied the effect of prestraining and cross-slip on the dislocation microstructure evolution within DDD simulations. Using the D2C method, we extracted continuous field descriptions of the dislocation microstructure that we used to quantify the similarity of the different initial conditions. Considering cross-slip yields dislocation microstructures that are more similar to each other and less dependent on the previous deformation history than if cross-slip was not taken into account. Including cross-slip is computationally more expensive due to higher dislocation densities. But the similarity of the resulting dislocation microstructures indicates that we can use fewer realizations to study statistical effects. This alleviates the higher computational cost of considering cross-slip.

Finally, we used the continuous fields extracted from DDD simulation results via the D2C method and successfully used them as inputs for machine learning model to classify the particle size of dislocation microstructures. Depending on whether the dislocation microstructure comprises mostly GNDs or SSDs, different combinations of dislocation density fields yielded better machine learning models. But a preference for higher-order fields was observed in both cases. The preferred spatial discretization also depended on the experiment, albeit it is less clear whether that is due to the GNDs and SSDs difference, or due to the different loading setup. We also introduced the dislocation feature density tensor that shows good potential as input for SSD-dominated dislocation microstructures.

Via the permutation feature importance, we were able to study in which way dislocation microstructures of different particle sizes differed. In the relaxation experiments that yielded configurations comprising predominantly SSDs, the most distinguishing features are found not close at the surface, but towards the center of the specimen. Thus, we were able to conclude that the dislocation microstructure close to a surface does not depend on the size of the specimen. This demonstrates how machine learning models can help us understand more about the formation dislocation microstructures and how they differ not only on average, but on an individual level.

Overall, we demonstrated that the D2C method is a powerful tool that enables us to use modern statistical approaches to study dislocation microstructures in great detail. The ability to quantitatively compare and average dislocation microstructures enables data-driven approaches that can provide insight into the effects of different dislocation reactions and deformation modes over a large number of samples. We now have all tools at hand that are required to gather a sufficient amount of data both from experiments and simulations to apply powerful machine learning techniques.

This is an important incentive to generate a community-wide database for dislocation microstructures. Simulation methods like DDD could benefit from experimental data that allows the developers to validate their frameworks using real data. In the other direction, experimentalists could take advantage of the additional information that can be provided by simulations. Consider a researcher who finds a peculiar characteristic in the dislocation microstructure he obtained form an experiment or a simulation. The researcher could then upload this dislocation microstructure and find the results of simulations that exhibit similar features. By comparing the simulation results to the experimental results, conclusion of the underlying mechanisms in the experiment may be revealed. In combination with a sufficiently large database that provides detailed metadata, previously unknown underlying relationships might be discovered using unsupervised learning techniques.

Another potential application is the acceleration of CDD simulations. Similarly to how Yang et al. [81] used deep learning and trained a neural network to accurately predict the elastic response of complex two-phase microstructures, we could train machine learning models to simulate the evolution of dislocation microstructures. As data-driven approaches tend to be computationally far less expensive than conventional FEM computations, we could study the behavior of far larger specimen. Additionally, by analyzing the information used by successful machine learning models, we might contribute to the development of CDD theories.

Appendix

A. Remaining relative permutation feature importances of the relaxation experiment

The remaining permutation feature importances of the indentation experiment of section 6.2 are shown in figures A.1 to A.9.

B. Remaining relative permutation feature importances of the indentation experiment

The remaining permutation feature importances of the indentation experiment of section 6.3 are shown in figures B.1 to B.6.

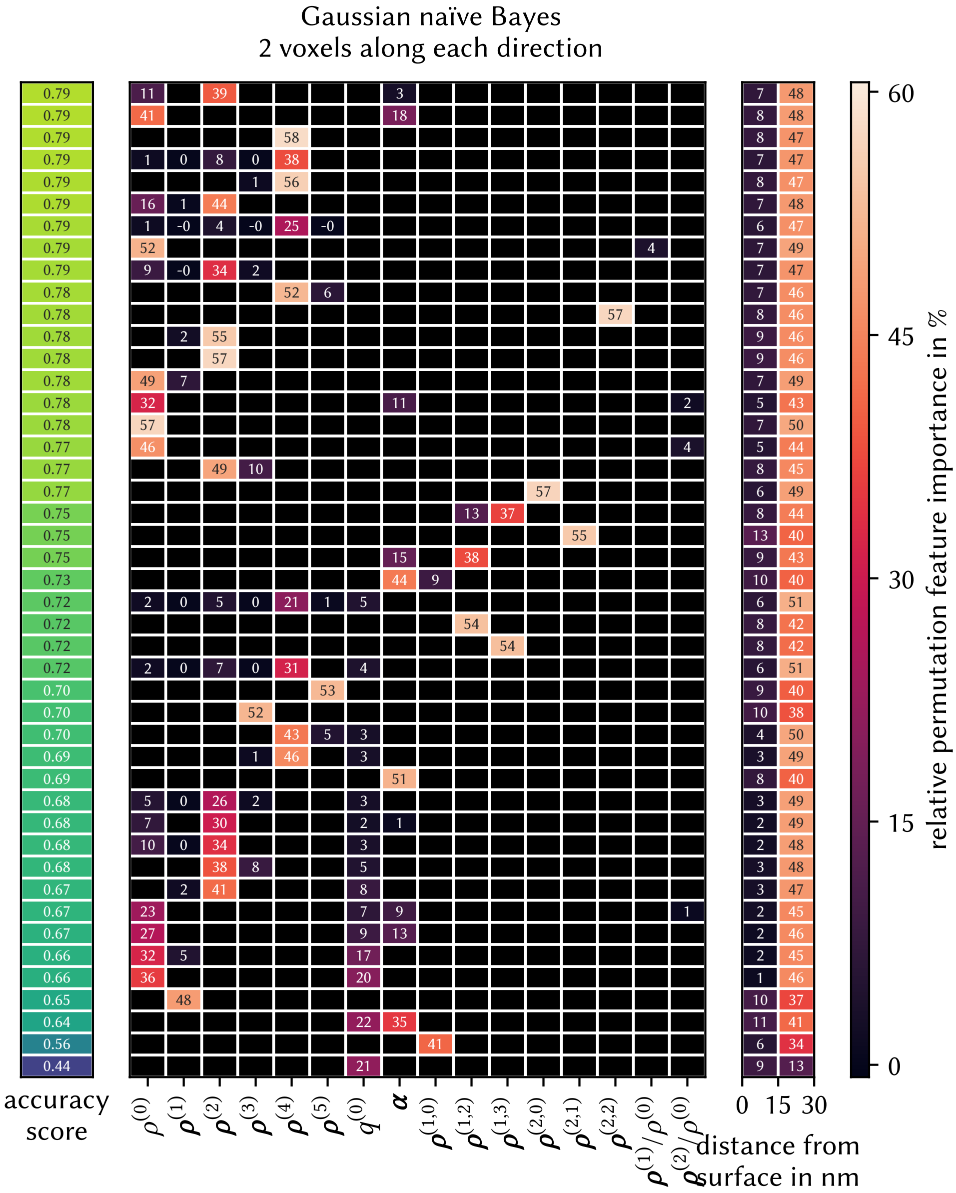


Figure A.1.: Relative permutation feature importance of each employed field and each distance from the surface of each voxel for all field combinations for 2 voxels along each direction for the GNB-based machine learning models. The field combinations are sorted by the achieved accuracy score.

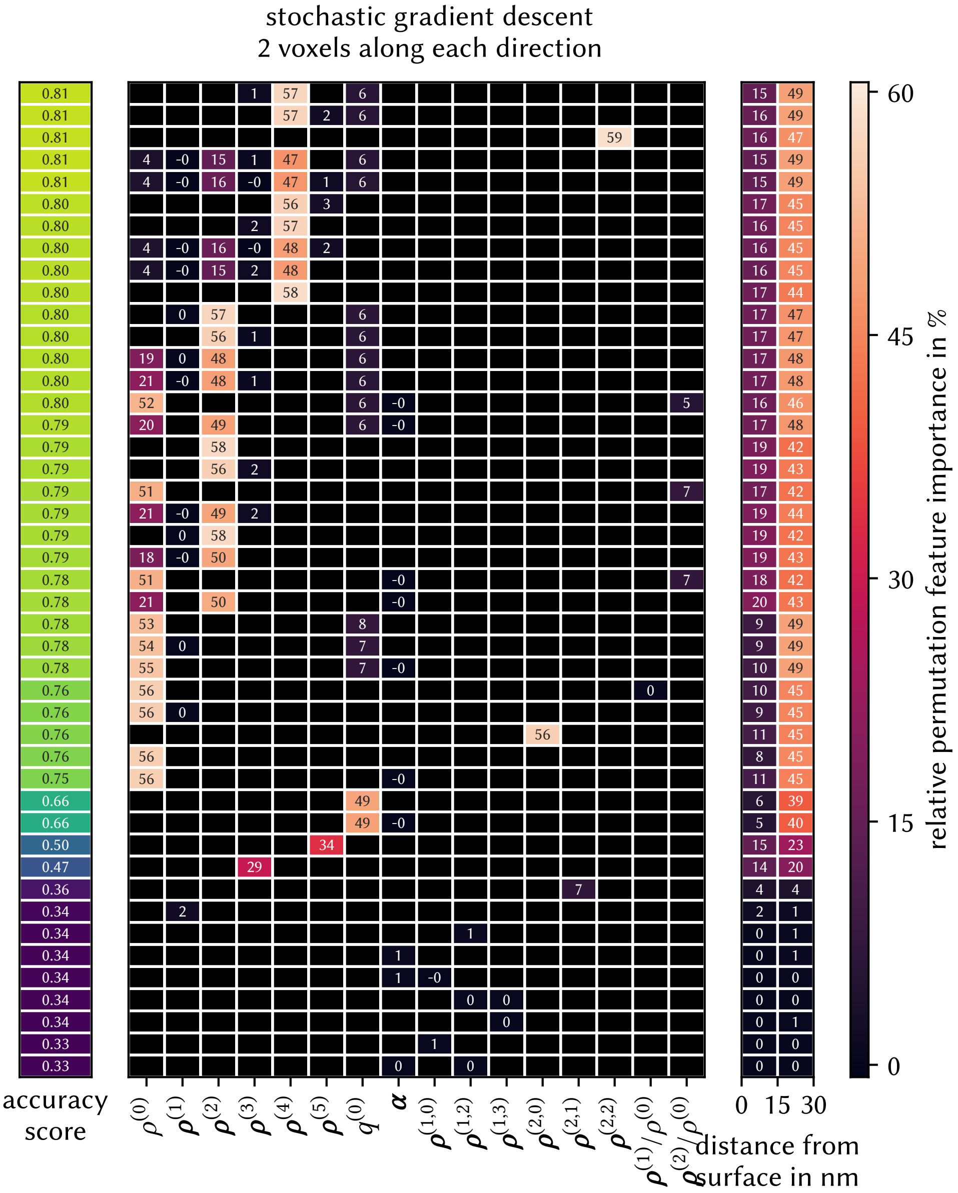


Figure A.2.: Relative permutation feature importance of each employed field and each distance from the surface of each voxel for all field combinations for 2 voxels along each direction for the SGD-based machine learning models. The field combinations are sorted by the achieved accuracy score.

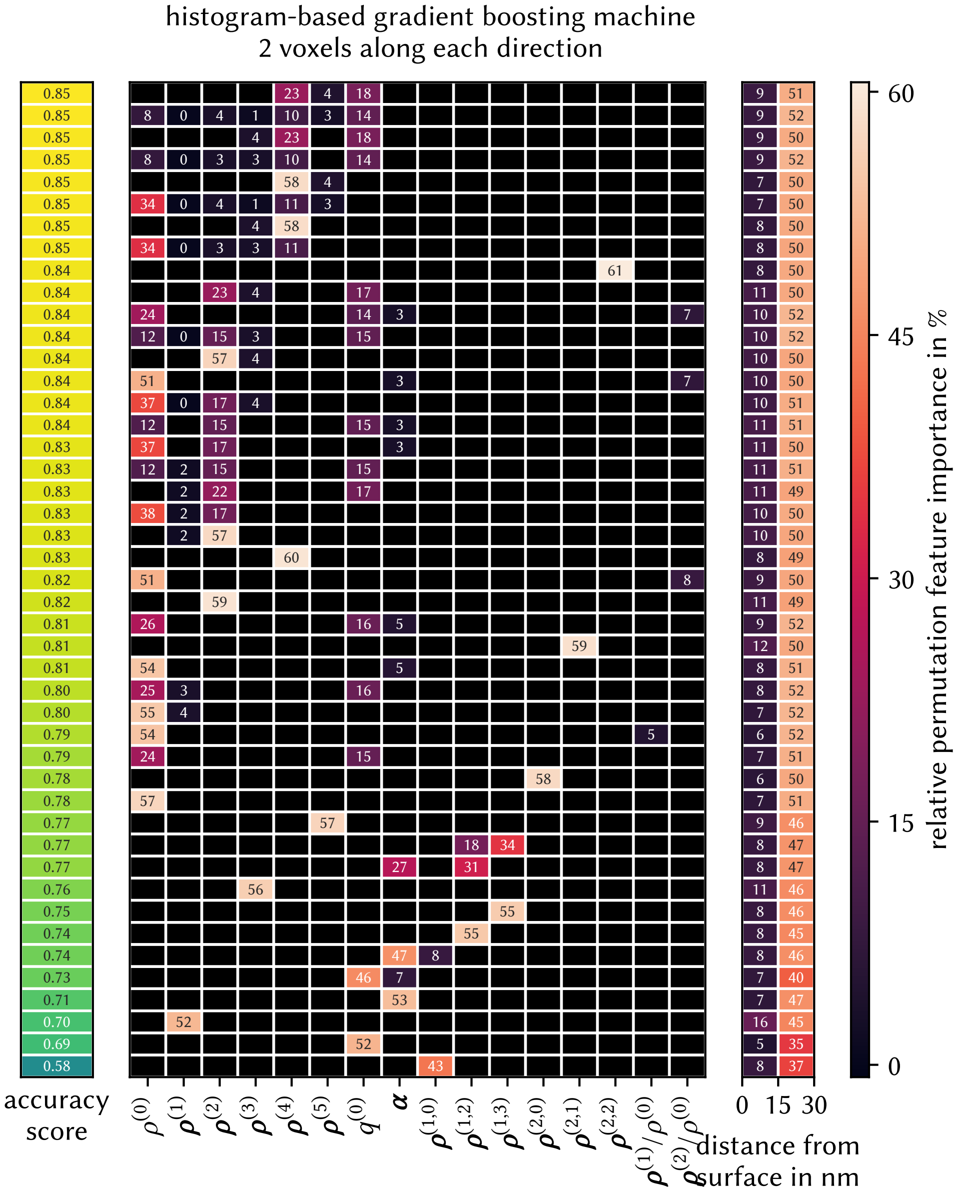


Figure A.3.: Relative permutation feature importance of each employed field and each distance from the surface of each voxel for all field combinations for 2 voxels along each direction for the HGBM-based machine learning models. The field combinations are sorted by the achieved accuracy score.

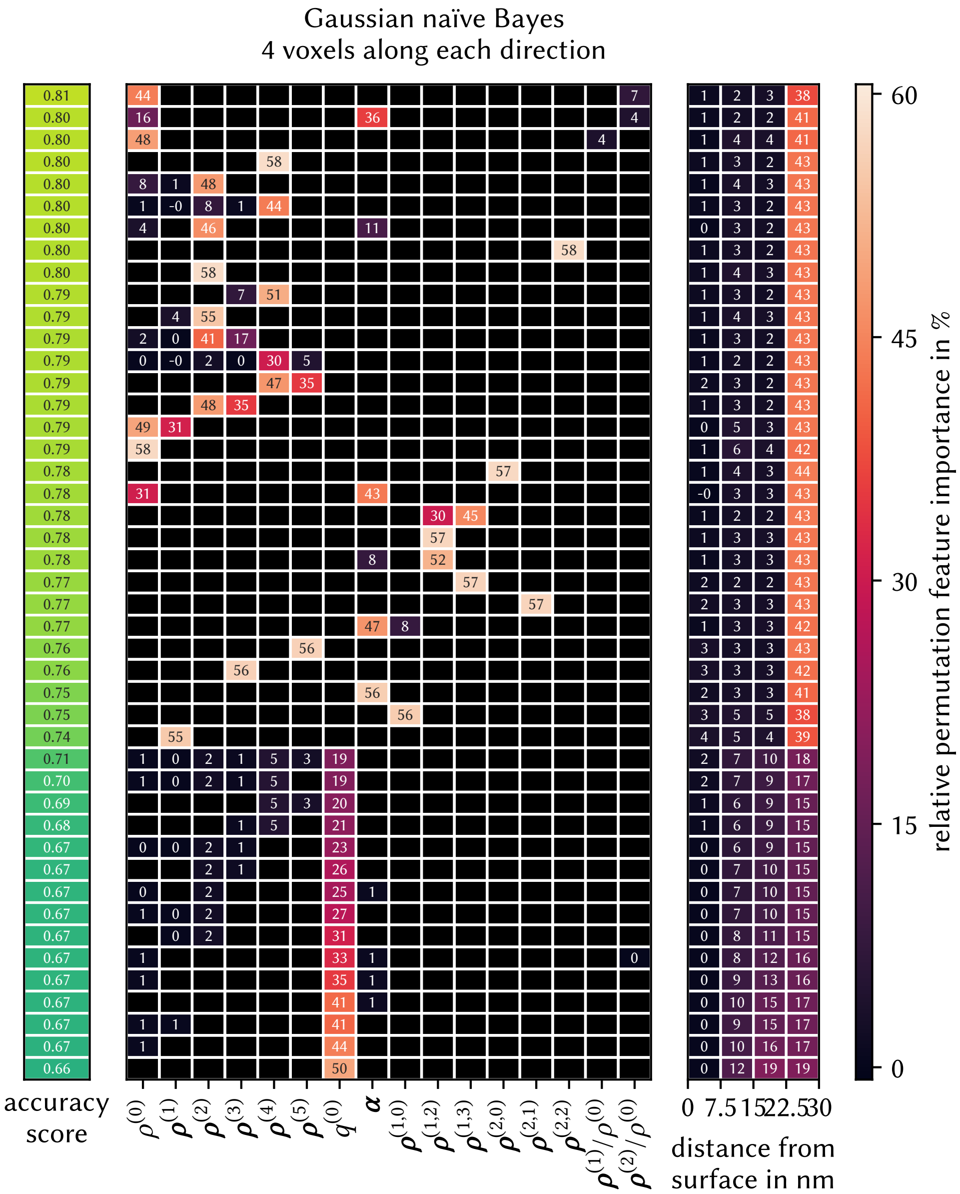


Figure A.4.: Relative permutation feature importance of each employed field and each distance from the surface of each voxel for all field combinations for 4 voxels along each direction for the GNB-based machine learning models. The field combinations are sorted by the achieved accuracy score.

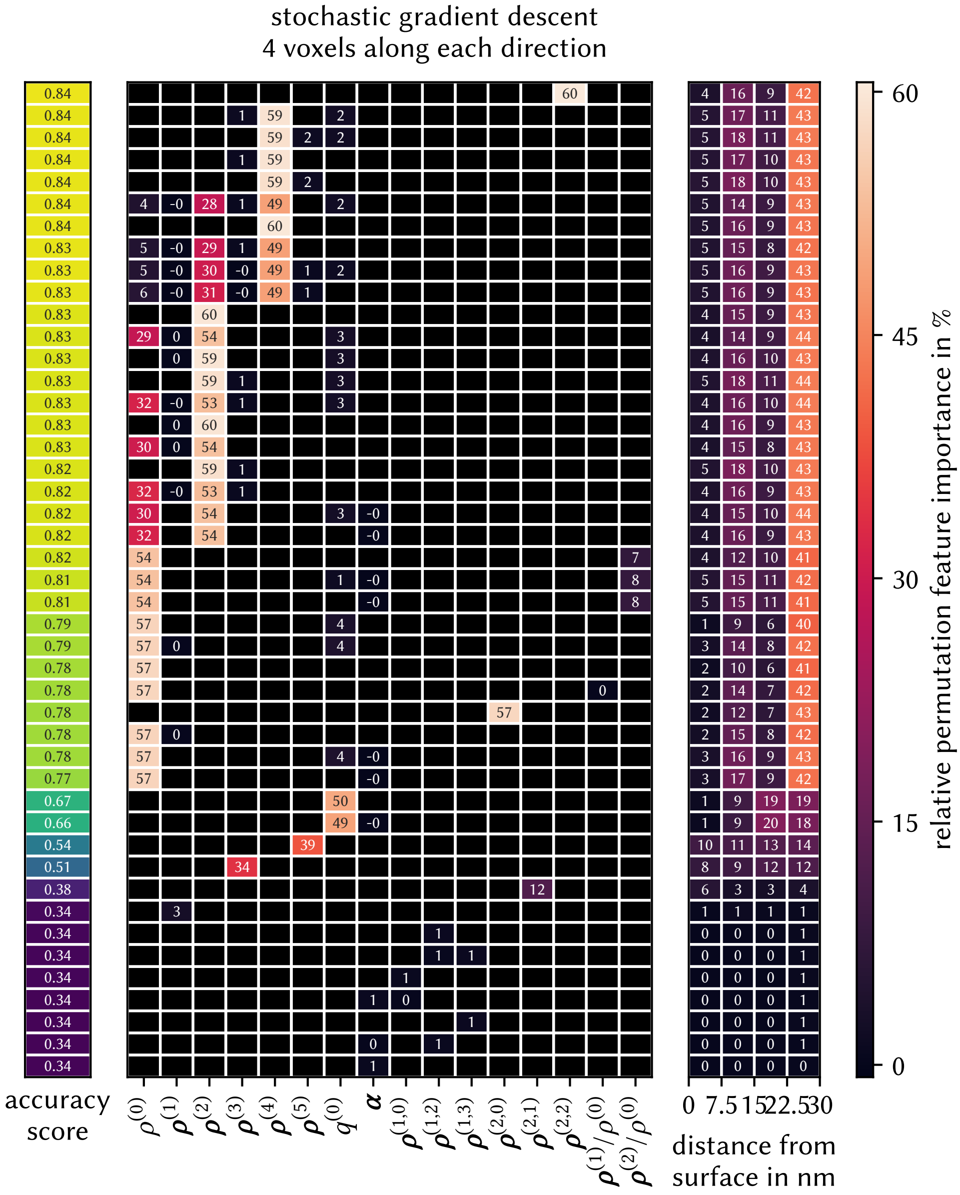
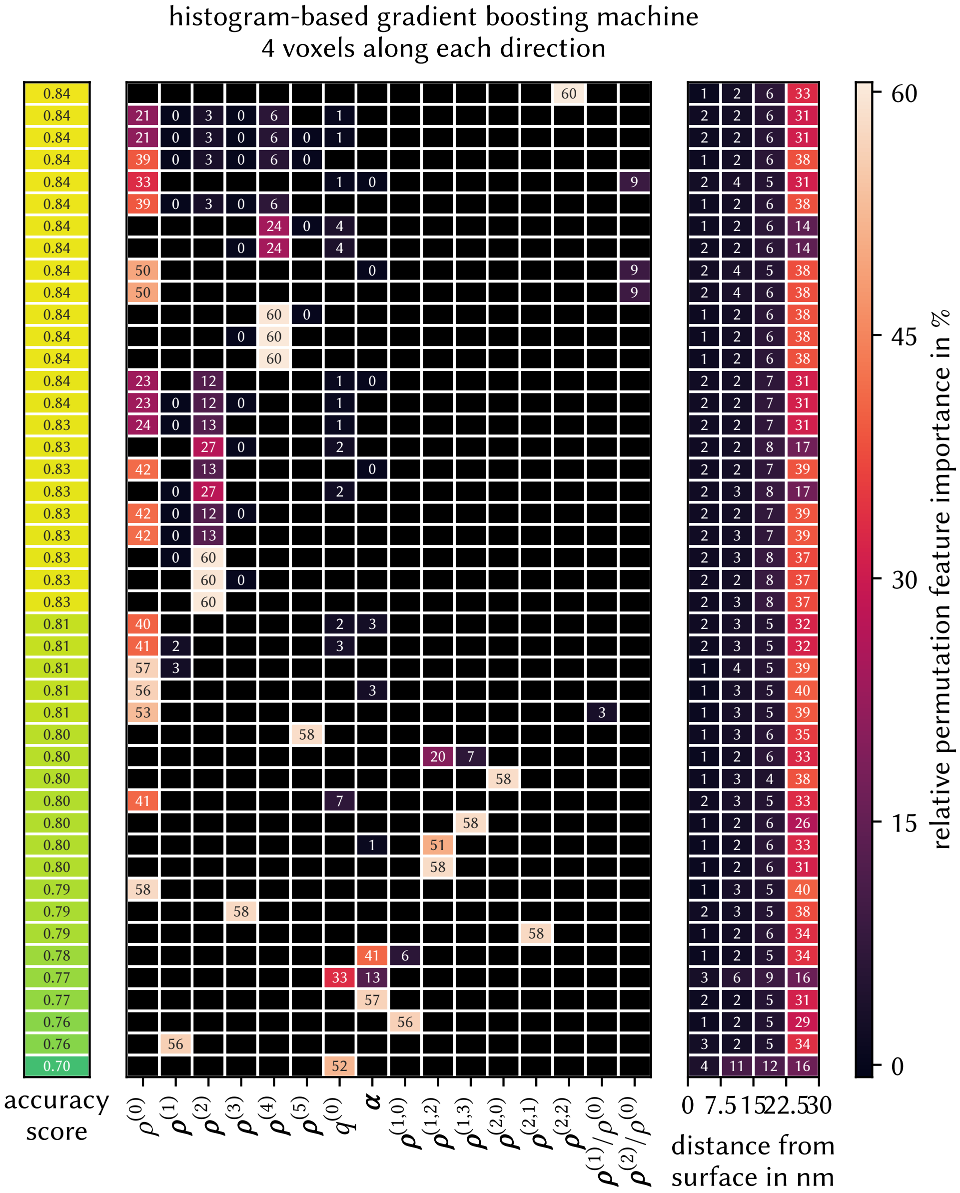


Figure A.5.: Relative permutation feature importance of each employed field and each distance from the surface of each voxel for all field combinations for 4 voxels along each direction for the SGD-based machine learning models. The field combinations are sorted by the achieved accuracy score.



accuracy score	$\rho^{(0)}$	$\boldsymbol{\rho}^{(1)}$	$\boldsymbol{\rho}^{(2)}$	$\boldsymbol{\rho}^{(3)}$	$\boldsymbol{\rho}^{(4)}$	$\boldsymbol{\rho}^{(5)}$	$q^{(0)}$	$\boldsymbol{\alpha}$	$\boldsymbol{\rho}^{(1,0)}$	$\boldsymbol{\rho}^{(1,2)}$	$\boldsymbol{\rho}^{(1,3)}$	$\boldsymbol{\rho}^{(2,0)}$	$\boldsymbol{\rho}^{(2,1)}$	$\boldsymbol{\rho}^{(2,2)}$	$\rho^{(1)}/\rho^{(0)}$	$\rho^{(2)}/\rho^{(0)}$				
0.84														60			1	2	6	33
0.84	21	0	3	0	6		1										2	2	6	31
0.84	21	0	3	0	6	0	1										2	2	6	31
0.84	39	0	3	0	6	0											1	2	6	38
0.84	33						1	0								9	2	4	5	31
0.84	39	0	3	0	6												1	2	6	38
0.84					24	0	4										1	2	6	14
0.84				0	24		4										2	2	6	14
0.84	50							0								9	2	4	5	38
0.84	50															9	2	4	6	38
0.84					60	0											1	2	6	38
0.84				0	60												1	2	6	38
0.84					60												1	2	6	38
0.84	23		12				1	0									2	2	7	31
0.84	23	0	12	0			1										2	2	7	31
0.83	24	0	13				1										2	2	7	31
0.83			27	0			2										2	2	8	17
0.83	42		13					0									2	2	7	39
0.83		0	27				2										2	3	8	17
0.83	42	0	12	0													2	2	7	39
0.83	42	0	13														2	3	7	39
0.83		0	60														2	3	8	37
0.83			60	0													2	2	8	37
0.83			60														2	3	8	37
0.81	40						2	3									2	3	5	32
0.81	41	2					3										2	3	5	32
0.81	57	3															1	4	5	39
0.81	56							3									1	3	5	40
0.81	53														3		1	3	5	39
0.80						58											1	3	6	35
0.80										20	7						1	2	6	33
0.80												58					1	3	4	38
0.80	41						7										2	3	5	33
0.80											58						1	2	6	26
0.80								1		51							1	2	6	33
0.80										58							1	2	6	31
0.79	58																1	3	5	40
0.79				58													2	3	5	38
0.79													58				1	2	6	34
0.78								41	6								1	2	5	34
0.77							33	13									3	6	9	16
0.77								57									2	2	5	31
0.76									56								1	2	5	29
0.76		56															3	2	5	34
0.70							52										4	11	12	16

Figure A.6.: Relative permutation feature importance of each employed field and each distance from the surface of each voxel for all field combinations for 4 voxels along each direction for the HGBM-based machine learning models. The field combinations are sorted by the achieved accuracy score.

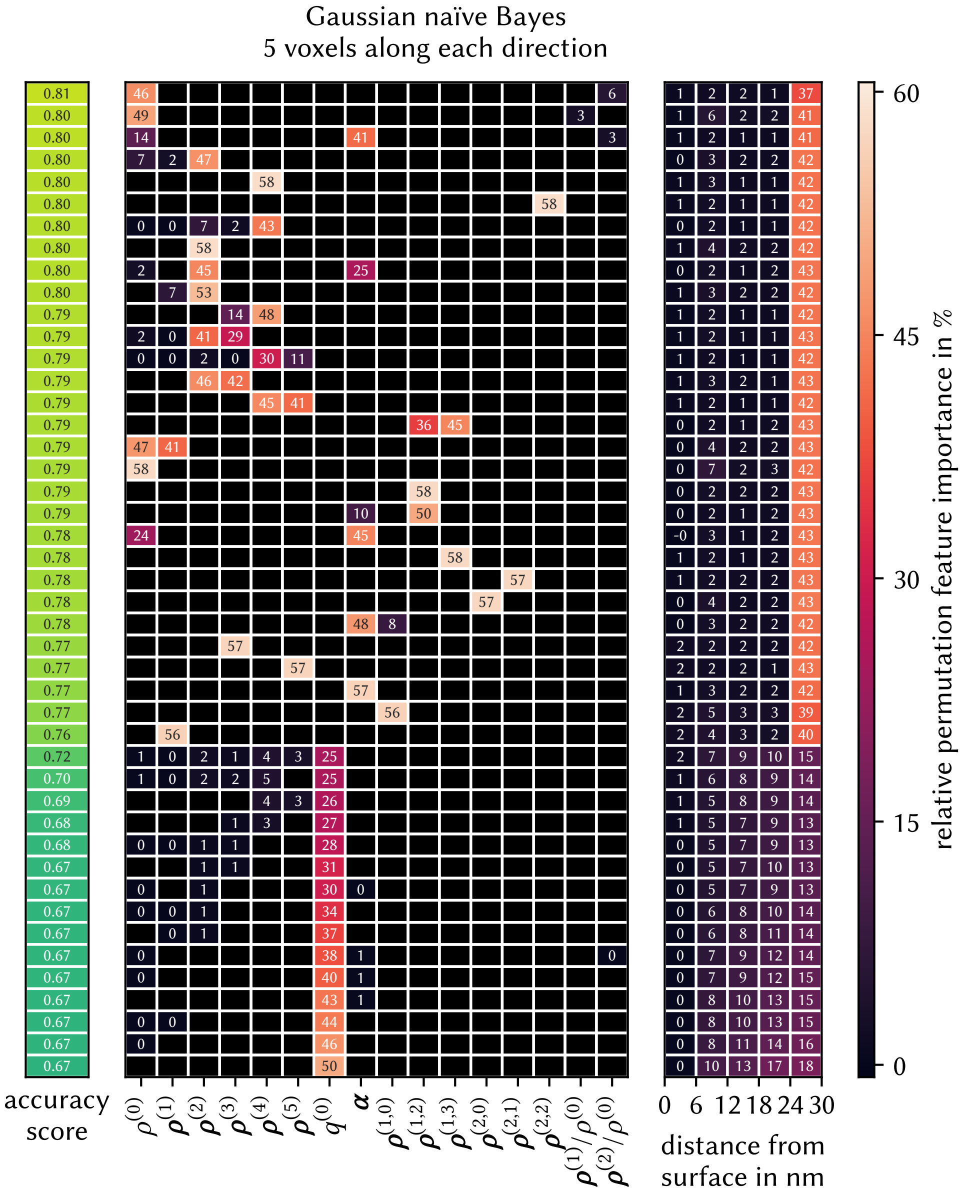
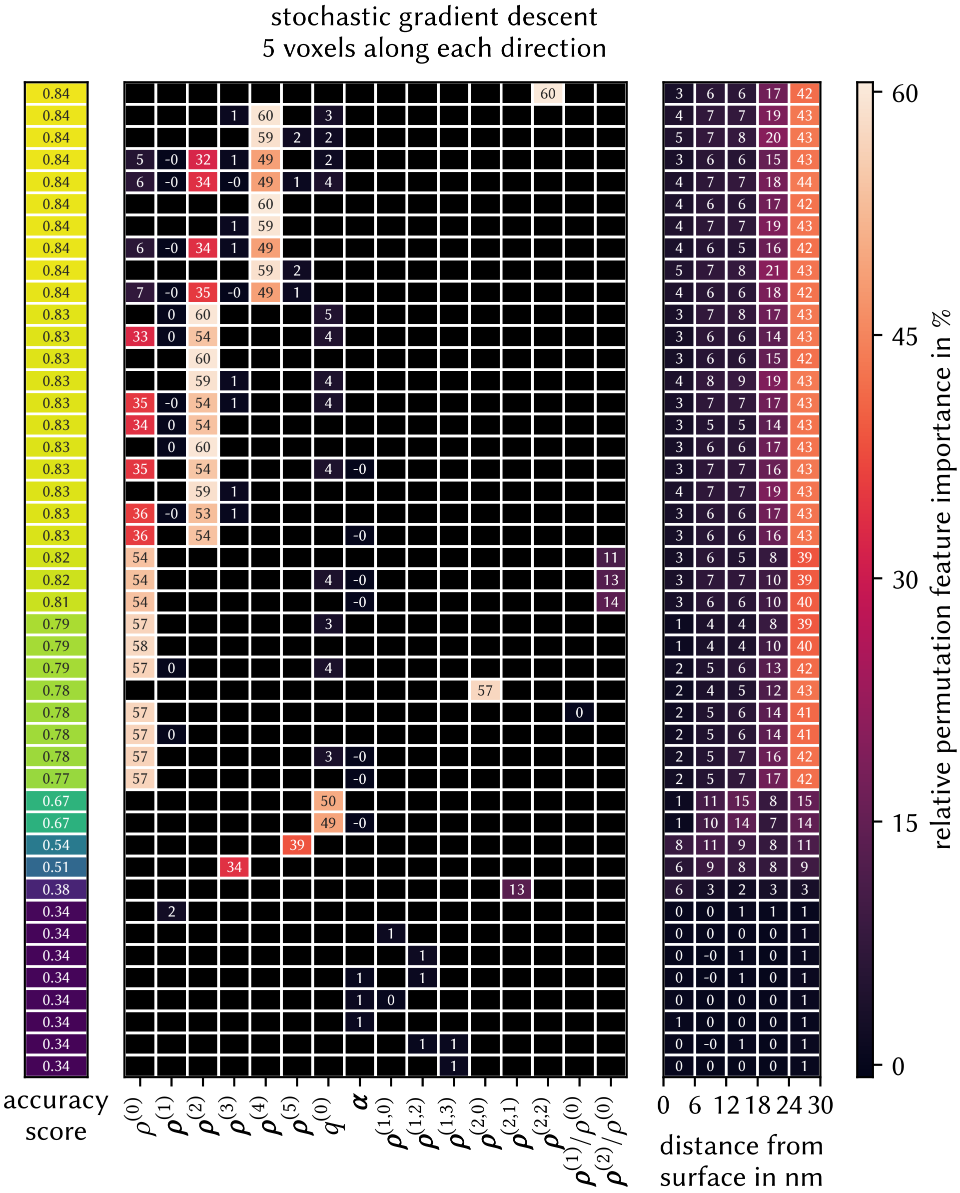


Figure A.7.: Relative permutation feature importance of each employed field and each distance from the surface of each voxel for all field combinations for 5 voxels along each direction for the GNB-based machine learning models. The field combinations are sorted by the achieved accuracy score.



accuracy score	$\rho^{(0)}$	$\boldsymbol{\rho}^{(1)}$	$\boldsymbol{\rho}^{(2)}$	$\boldsymbol{\rho}^{(3)}$	$\boldsymbol{\rho}^{(4)}$	$\boldsymbol{\rho}^{(5)}$	$q^{(0)}$	$\boldsymbol{\alpha}$	$\boldsymbol{\rho}^{(1,0)}$	$\boldsymbol{\rho}^{(1,2)}$	$\boldsymbol{\rho}^{(1,3)}$	$\boldsymbol{\rho}^{(2,0)}$	$\boldsymbol{\rho}^{(2,1)}$	$\boldsymbol{\rho}^{(2,2)}$	$\boldsymbol{\rho}^{(1)}/\rho^{(0)}$	$\boldsymbol{\rho}^{(2)}/\rho^{(0)}$	0	6	12	18	24	30
0.84														60			3	6	6	17	42	
0.84				1	60		3										4	7	7	19	43	
0.84					59	2	2										5	7	8	20	43	
0.84	5	-0	32	1	49		2										3	6	6	15	43	
0.84	6	-0	34	-0	49	1	4										4	7	7	18	44	
0.84					60												4	6	6	17	42	
0.84				1	59												4	7	7	19	43	
0.84	6	-0	34	1	49												4	6	5	16	42	
0.84					59	2											5	7	8	21	43	
0.84	7	-0	35	-0	49	1											4	6	6	18	42	
0.83		0	60				5										3	7	8	17	43	
0.83	33	0	54				4										3	6	6	14	43	
0.83			60														3	6	6	15	42	
0.83			59	1			4										4	8	9	19	43	
0.83	35	-0	54	1			4										3	7	7	17	43	
0.83	34	0	54														3	5	5	14	43	
0.83		0	60														3	6	6	17	43	
0.83	35		54				4	-0									3	7	7	16	43	
0.83			59	1													4	7	7	19	43	
0.83	36	-0	53	1													3	6	6	17	43	
0.83	36		54					-0									3	6	6	16	43	
0.82	54															11	3	6	5	8	39	
0.82	54						4	-0								13	3	7	7	10	39	
0.81	54							-0								14	3	6	6	10	40	
0.79	57						3										1	4	4	8	39	
0.79	58																1	4	4	10	40	
0.79	57	0					4										2	5	6	13	42	
0.78												57					2	4	5	12	43	
0.78	57														0		2	5	6	14	41	
0.78	57	0															2	5	6	14	41	
0.78	57						3	-0									2	5	7	16	42	
0.77	57							-0									2	5	7	17	42	
0.67							50										1	11	15	8	15	
0.67							49	-0									1	10	14	7	14	
0.54						39											8	11	9	8	11	
0.51				34													6	9	8	8	9	
0.38													13				6	3	2	3	3	
0.34		2															0	0	1	1	1	
0.34									1								0	0	0	0	1	
0.34										1							0	-0	1	0	1	
0.34								1		1							0	-0	1	0	1	
0.34								1	0								0	0	0	0	1	
0.34								1									1	0	0	0	1	
0.34										1	1						0	-0	1	0	1	
0.34											1						0	0	0	0	1	

Figure A.8.: Relative permutation feature importance of each employed field and each distance from the surface of each voxel for all field combinations for 5 voxels along each direction for the SGD-based machine learning models. The field combinations are sorted by the achieved accuracy score.

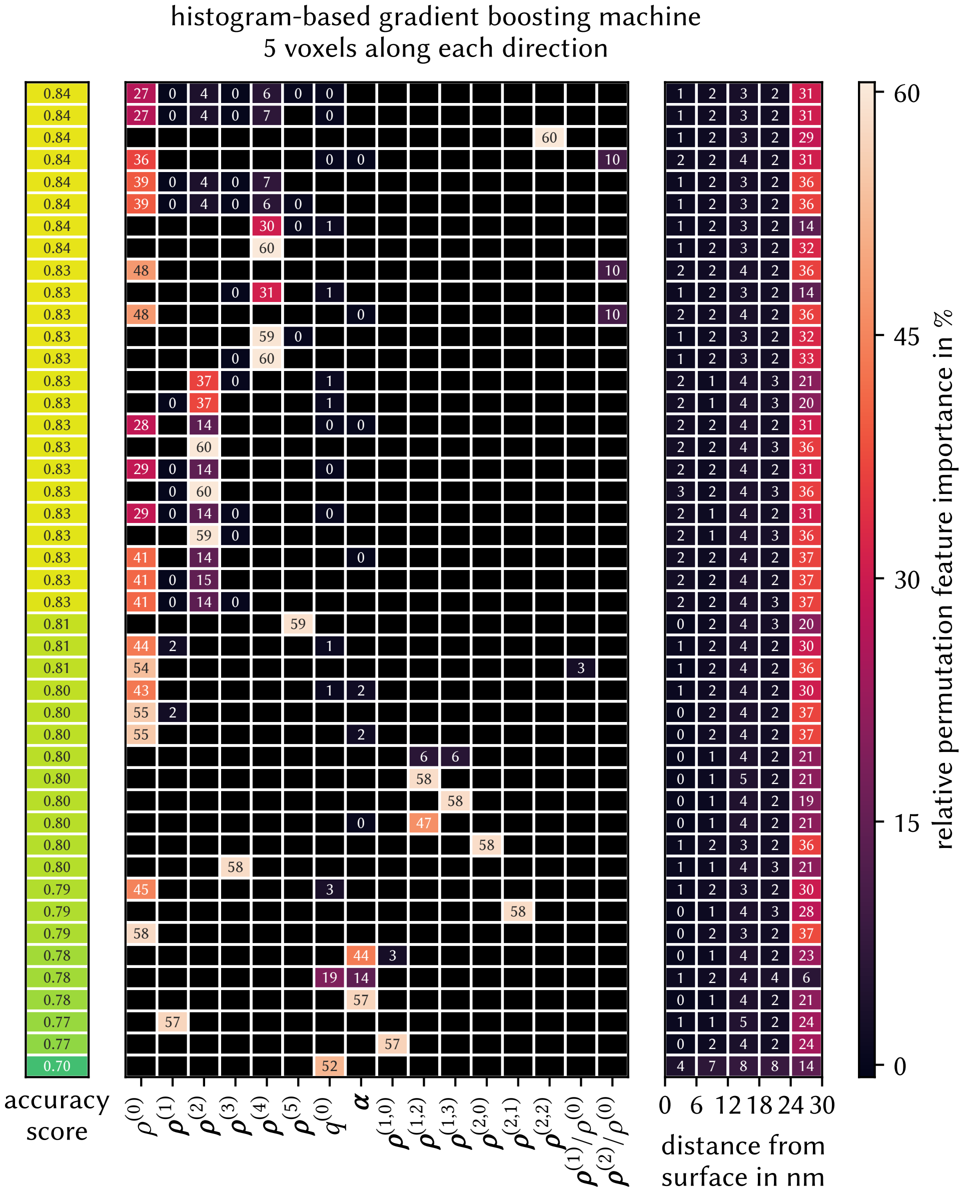


Figure A.9.: Relative permutation feature importance of each employed field and each distance from the surface of each voxel for all field combinations for 5 voxels along each direction for the HGBM-based machine learning models. The field combinations are sorted by the achieved accuracy score.

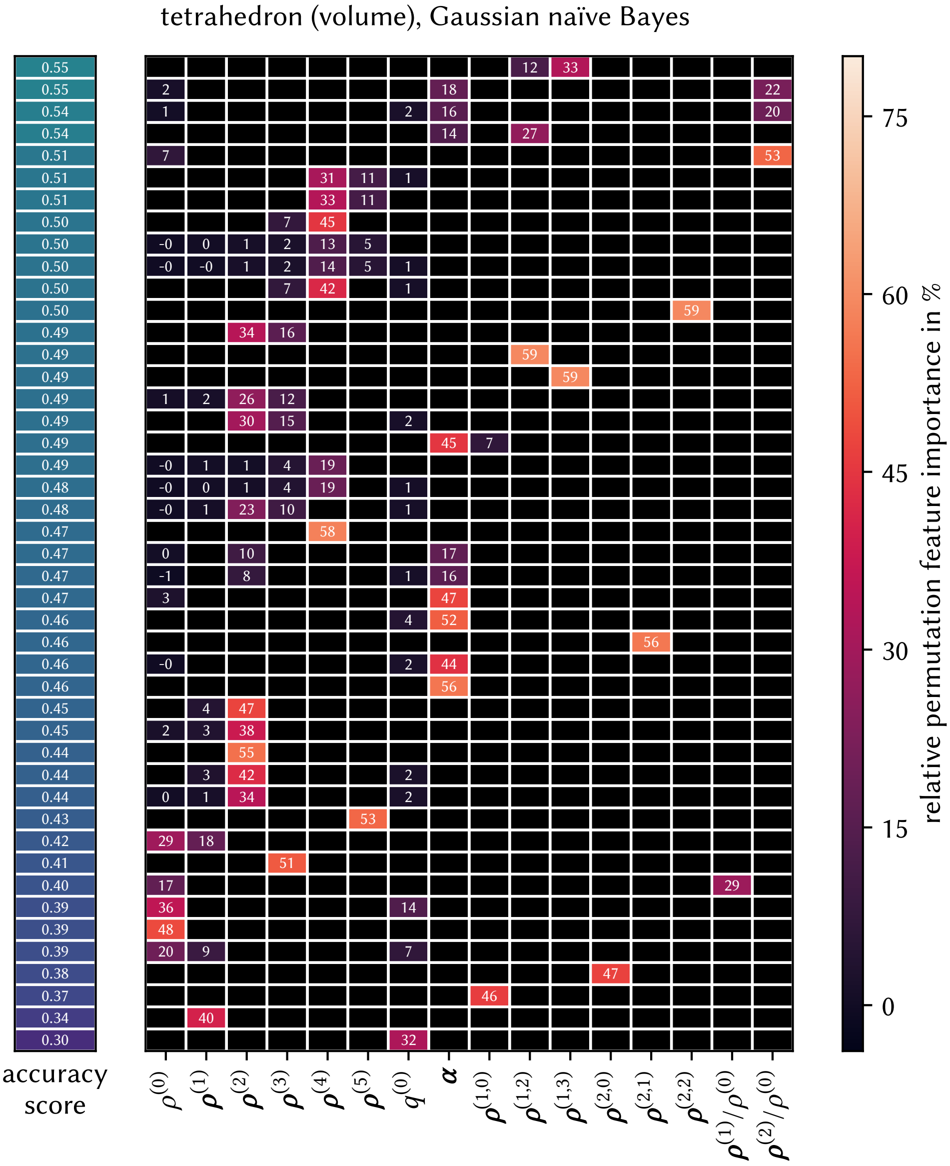
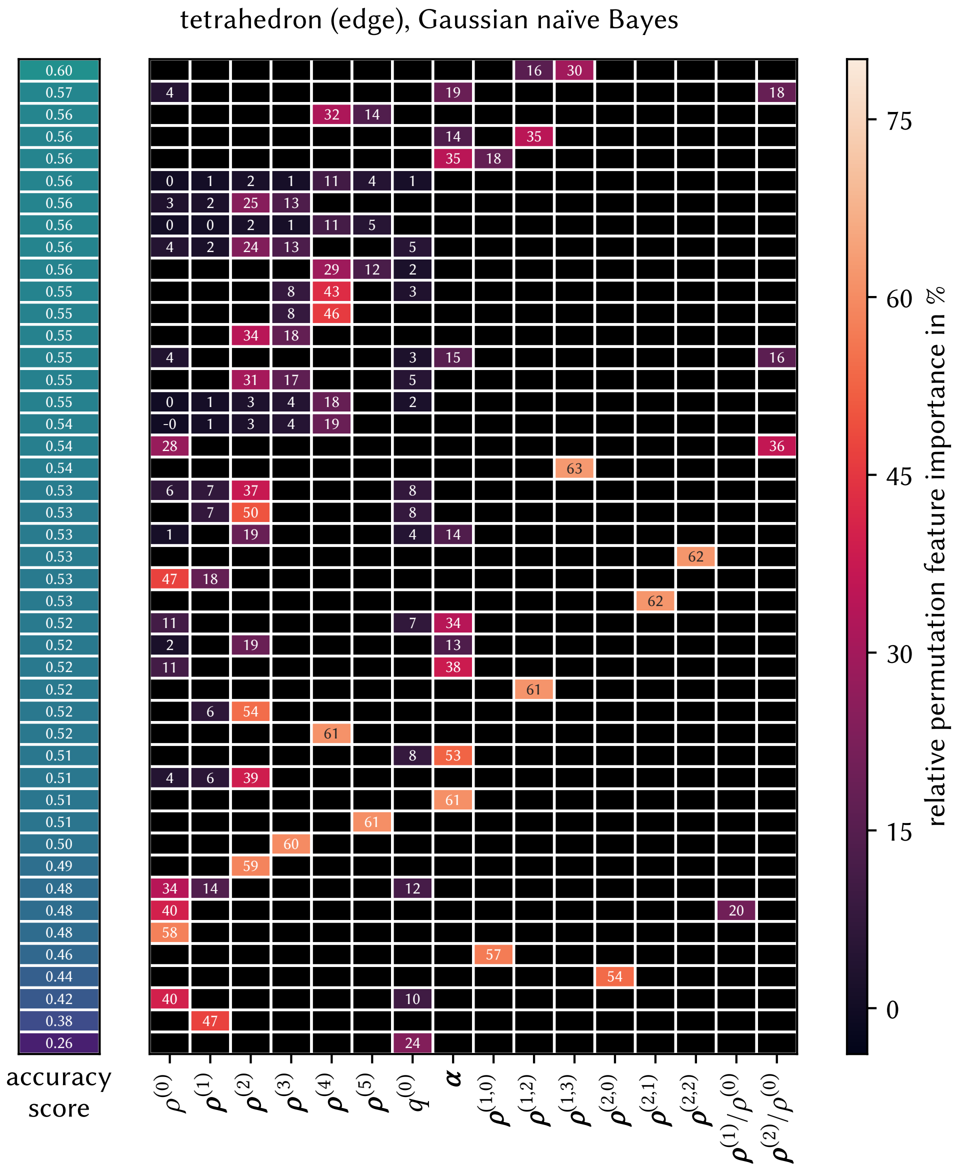
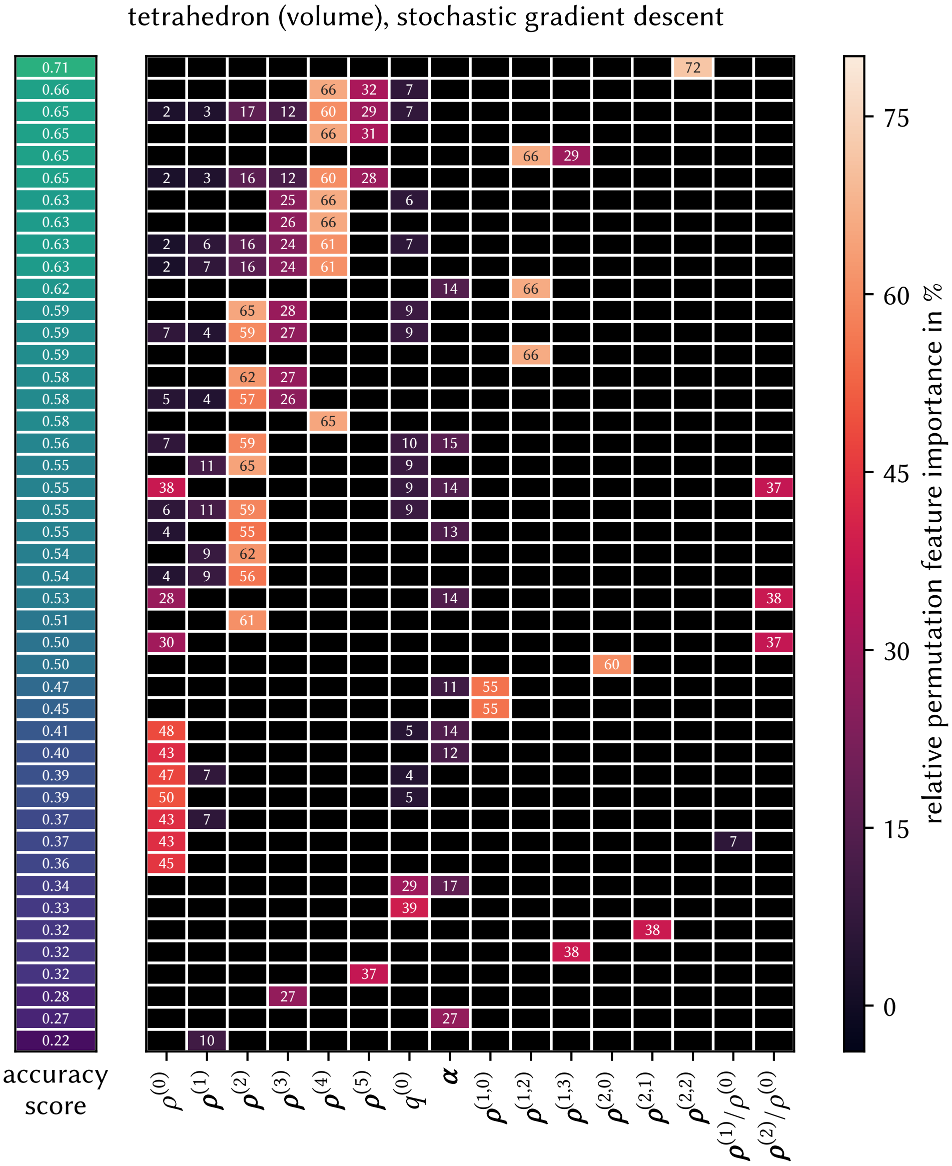


Figure B.1.: Relative permutation feature importance of each employed field for all field combinations for the tetrahedron-like (volume) spatial discretization for the GNB-based machine learning models. The values are averages for all rotations of the spatial discretization. The field combinations are sorted by the achieved accuracy score.



accuracy score	$\rho^{(0)}$	$\rho^{(1)}$	$\rho^{(2)}$	$\rho^{(3)}$	$\rho^{(4)}$	$\rho^{(5)}$	$q^{(0)}$	α	$\rho^{(1,0)}$	$\rho^{(1,2)}$	$\rho^{(1,3)}$	$\rho^{(2,0)}$	$\rho^{(2,1)}$	$\rho^{(2,2)}$	$\rho^{(1)}/\rho^{(0)}$	$\rho^{(2)}/\rho^{(0)}$
0.60										16	30					
0.57	4							19								18
0.56					32	14										
0.56								14		35						
0.56								35	18							
0.56	0	1	2	1	11	4	1									
0.56	3	2	25	13												
0.56	0	0	2	1	11	5										
0.56	4	2	24	13			5									
0.56					29	12	2									
0.55				8	43		3									
0.55				8	46											
0.55			34	18												
0.55	4						3	15								16
0.55			31	17			5									
0.55	0	1	3	4	18		2									
0.54	-0	1	3	4	19											
0.54	28															36
0.54											63					
0.53	6	7	37				8									
0.53		7	50				8									
0.53	1		19				4	14								
0.53														62		
0.53	47	18														
0.53													62			
0.52	11						7	34								
0.52	2		19					13								
0.52	11							38								
0.52										61						
0.52		6	54													
0.52					61											
0.51							8	53								
0.51	4	6	39													
0.51								61								
0.51						61										
0.50				60												
0.49			59													
0.48	34	14					12									
0.48	40														20	
0.48	58															
0.46									57							
0.44												54				
0.42	40						10									
0.38		47														
0.26							24									

Figure B.2.: Relative permutation feature importance of each employed field for all field combinations for the tetrahedron-like (edge) spatial discretization for the GNB-based machine learning models. The values are averages for all rotations of the spatial discretization. The field combinations are sorted by the achieved accuracy score.



accuracy score	$\rho^{(0)}$	$\rho^{(1)}$	$\rho^{(2)}$	$\rho^{(3)}$	$\rho^{(4)}$	$\rho^{(5)}$	$q^{(0)}$	α	$\rho^{(1,0)}$	$\rho^{(1,2)}$	$\rho^{(1,3)}$	$\rho^{(2,0)}$	$\rho^{(2,1)}$	$\rho^{(2,2)}$	$\rho^{(1)}/\rho^{(0)}$	$\rho^{(2)}/\rho^{(0)}$
0.71														72		
0.66					66	32	7									
0.65	2	3	17	12	60	29	7									
0.65					66	31										
0.65										66	29					
0.65	2	3	16	12	60	28										
0.63				25	66		6									
0.63				26	66											
0.63	2	6	16	24	61		7									
0.63	2	7	16	24	61											
0.62								14		66						
0.59			65	28			9									
0.59	7	4	59	27			9									
0.59										66						
0.58			62	27												
0.58	5	4	57	26												
0.58					65											
0.56	7		59				10	15								
0.55		11	65				9									
0.55	38						9	14								37
0.55	6	11	59				9									
0.55	4		55					13								
0.54		9	62													
0.54	4	9	56													
0.53	28							14								38
0.51			61													
0.50	30															37
0.50												60				
0.47								11	55							
0.45									55							
0.41	48						5	14								
0.40	43							12								
0.39	47	7					4									
0.39	50						5									
0.37	43	7														
0.37	43														7	
0.36	45															
0.34							29	17								
0.33							39									
0.32													38			
0.32											38					
0.32						37										
0.28				27												
0.27								27								
0.22		10														

Figure B.3.: Relative permutation feature importance of each employed field for all field combinations for the tetrahedron-like (volume) spatial discretization for the SGD-based machine learning models. The values are averages for all rotations of the spatial discretization. The field combinations are sorted by the achieved accuracy score.

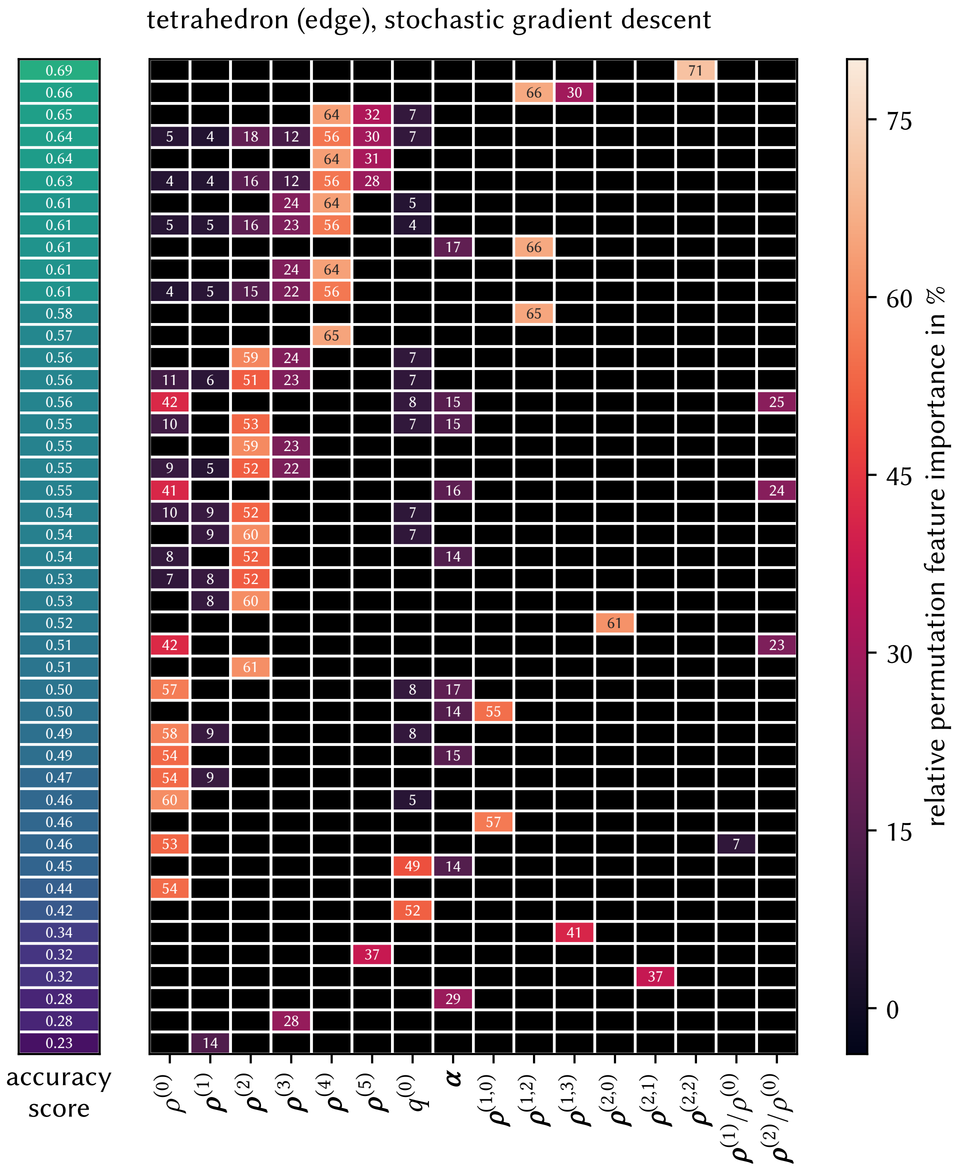


Figure B.4.: Relative permutation feature importance of each employed field for all field combinations for the tetrahedron-like (edge) spatial discretization for the SGD-based machine learning models. The values are averages for all rotations of the spatial discretization. The field combinations are sorted by the achieved accuracy score.

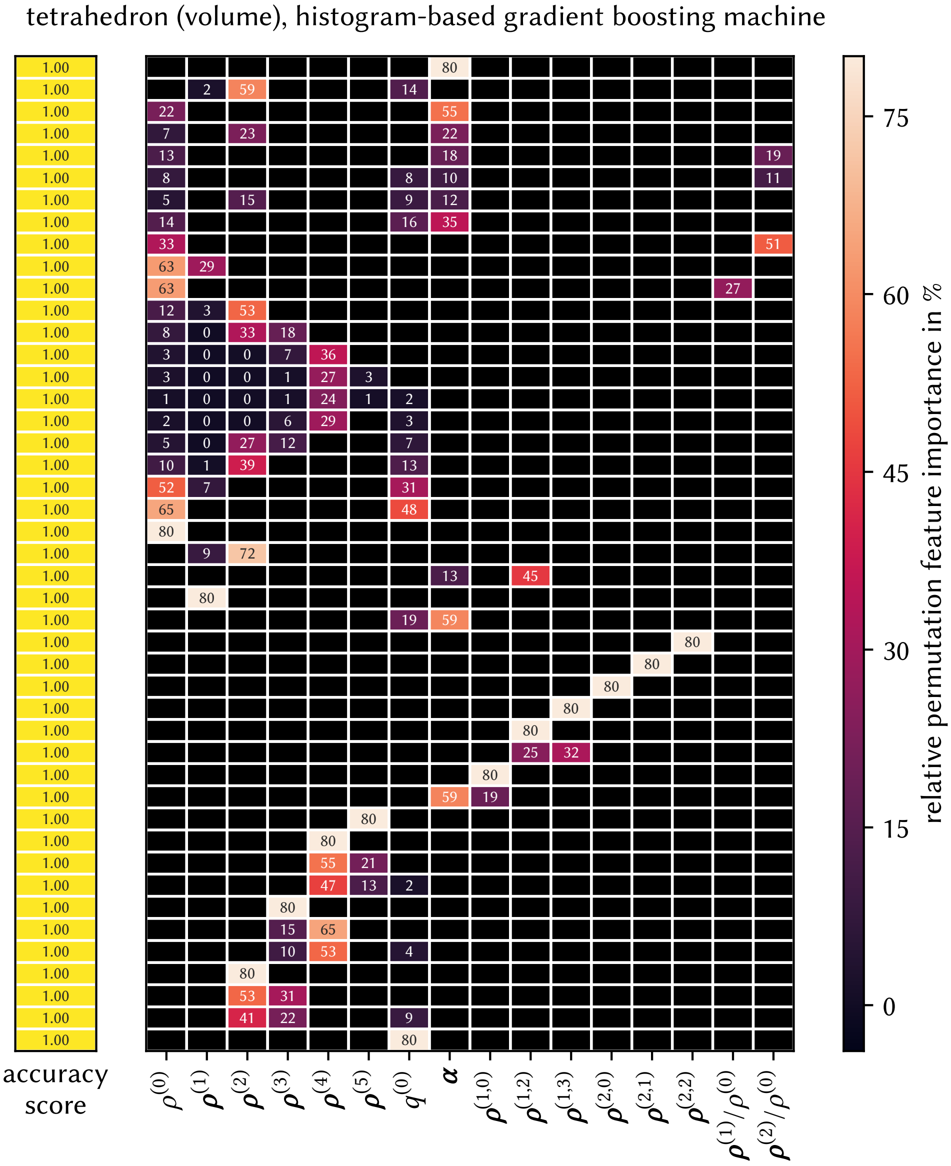


Figure B.5.: Relative permutation feature importance of each employed field for all field combinations for the tetrahedron-like (volume) spatial discretization for the HGBM-based machine learning models. The values are averages for all rotations of the spatial discretization. The field combinations are sorted by the achieved accuracy score.

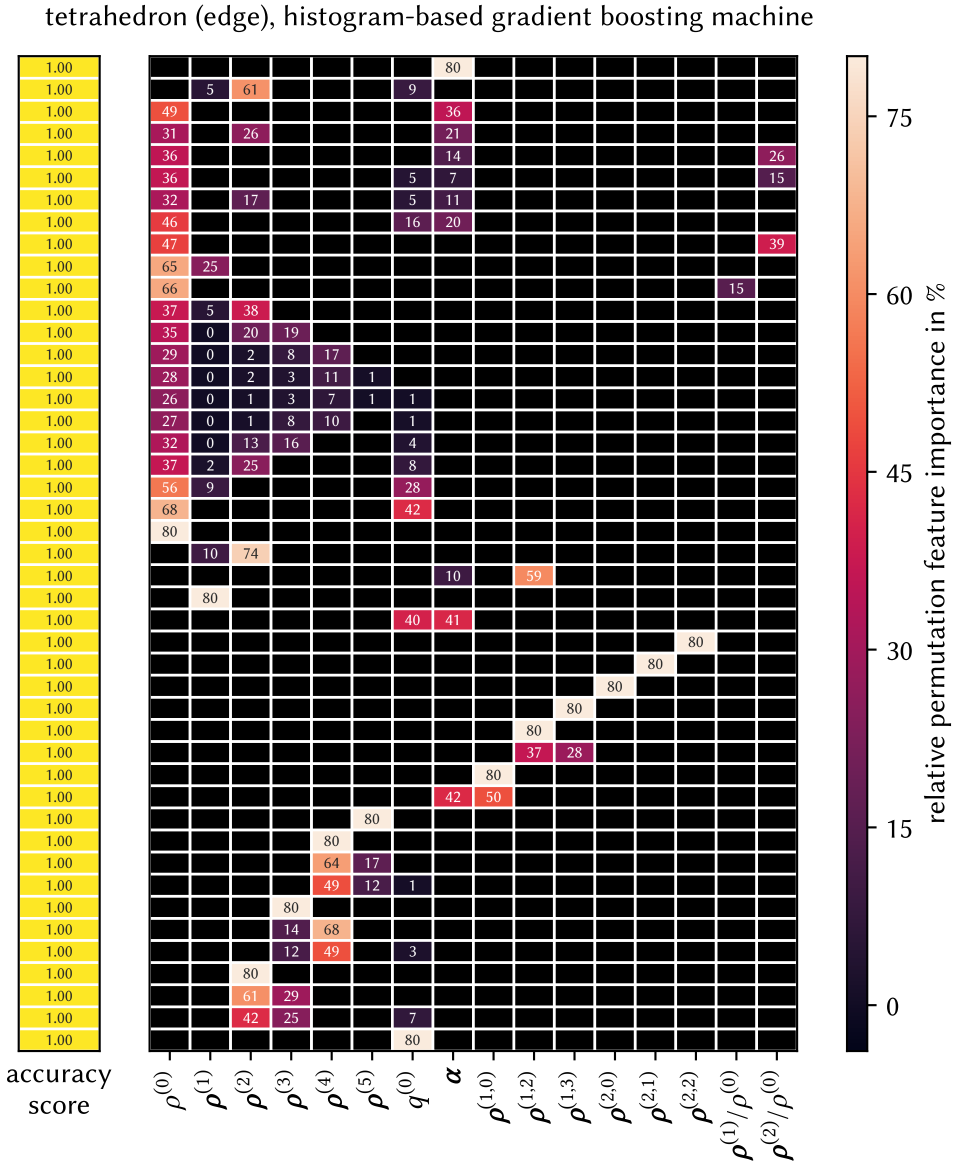


Figure B.6.: Relative permutation feature importance of each employed field for all field combinations for the tetrahedron-like (edge) spatial discretization for the HGBM-based machine learning models. The values are averages for all rotations of the spatial discretization. The field combinations are sorted by the achieved accuracy score.

Bibliography

[1] W. Cai et al. “A non-singular continuum theory of dislocations.” In: *Journal of the Mechanics and Physics of Solids* 54.3 (Mar. 2006), pp. 561–587.

[2] G. I. Taylor and C. F. Elam. “Bakerian Lecture: The distortion of an aluminium crystal during a tensile test.” In: *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 102.719 (Mar. 1923), pp. 643–667.

[3] E. Orowan. “Zur Kristallplastizität. I.” In: *Zeitschrift für Physik* 89.9-10 (Sept. 1934), pp. 605–613.

[4] E. Orowan. “Zur Kristallplastizität. III.” In: *Zeitschrift für Physik* 89.9-10 (Sept. 1934), pp. 634–659.

[5] M. Polanyi. “Über eine Art Gitterstörung, die einen Kristall plastisch machen könnte.” In: *Zeitschrift für Physik* 89.9-10 (Sept. 1934), pp. 660–664.

[6] G. I. Taylor. “The mechanism of plastic deformation of crystals. Part I.—Theoretical.” In: *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 145.855 (July 1934), pp. 362–387.

[7] J. W. Menter. “The direct study by electron microscopy of crystal lattices and their imperfections.” In: *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 236.1204 (July 10, 1956), pp. 119–135.

[8] P. B. Hirsch, R. W. Horne, and M. J. Whelan. “LXVIII. Direct observations of the arrangement and motion of dislocations in aluminium.” In: *Philosophical Magazine* 1.7 (July 1956), pp. 677–684.

[9] Z. Basinski, A. Korbel, and S. Basinski. “The temperature dependence of the saturation stress and dislocation substructure in fatigued copper single crystals.” In: *Acta Metallurgica* 28.2 (Feb. 1980), pp. 191–207.

[10] H. Mughrabi. “The long-range internal stress field in the dislocation wall structure of persistent slip bands.” In: *Physica Status Solidi (a)* 104.1 (Nov. 1987), pp. 107–120.

[11] H. Mughrabi. “The cyclic hardening and saturation behaviour of copper single crystals.” In: *Materials Science and Engineering* 33.2 (May 1978), pp. 207–223.

[12] N. Thompson, N. Wadsworth, and N. Louat. “Xi. The origin of fatigue fracture in copper.” In: *Philosophical Magazine* 1.2 (Feb. 1956), pp. 113–126.

[13] F. Mompiou et al. “In situ TEM observations of reverse dislocation motion upon unloading in tensile-deformed UFG aluminium.” In: *Acta Materialia* 60.8 (May 2012), pp. 3402–3414.

[14] C.-C. Chen et al. “Three-dimensional imaging of dislocations in a nanoparticle at atomic resolution.” In: *Nature* 496.7443 (Mar. 2013), pp. 74–77.

[15] S. Yamasaki et al. “3D visualization of dislocation arrangement using scanning electron microscope serial sectioning method.” In: *Scripta Materialia* 101 (May 2015), pp. 80–83.

[16] S. I. Rao et al. “Large-scale dislocation dynamics simulations of strain hardening of Ni microcrystals under tensile loading.” In: *Acta Materialia* 164 (Feb. 2019), pp. 171–183.

[17] T. Mura. “Continuous distribution of moving dislocations.” In: *Philosophical Magazine* 8.89 (May 1963), pp. 843–857.

[18] A. Acharya. “A model of crystal plasticity based on the theory of continuously distributed dislocations.” In: *Journal of the Mechanics and Physics of Solids* 49.4 (Apr. 2001), pp. 761–784.

[19] A. Acharya. “Driving forces and boundary conditions in continuum dislocation mechanics.” In: *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 459.2034 (June 2003), pp. 1343–1363.

[20] A. Arsenlis. “On the evolution of crystallographic dislocation density in non-homogeneously deforming crystals.” In: *Journal of the Mechanics and Physics of Solids* 52.6 (June 2004), pp. 1213–1246.

[21] T. Hochrainer, M. Zaiser, and P. Gumbsch. “A three-dimensional continuum theory of dislocation systems: kinematics and mean-field formulation.” In: *Philosophical Magazine* 87.8-9 (Mar. 2007), pp. 1261–1282.

[22] T. Hochrainer. “Multipole expansion of continuum dislocations dynamics in terms of alignment tensors.” In: *Philosophical Magazine* 95.12 (Apr. 2015), pp. 1321–1367.

[23] M. Monavari, S. Sandfeld, and M. Zaiser. “Continuum representation of systems of dislocation lines: A general method for deriving closed-form evolution equations.” In: *Journal of the Mechanics and Physics of Solids* 95 (Oct. 2016), pp. 575–601.

[24] A. Cecen, Y. C. Yabansu, and S. R. Kalidindi. “A new framework for rotationally invariant two-point spatial correlations in microstructure datasets.” In: *Acta Materialia* 158 (Oct. 2018), pp. 53–64.

[25] A. Chowdhury et al. “Image driven machine learning methods for microstructure recognition.” In: *Computational Materials Science* 123 (Oct. 2016), pp. 176–187.

[26] R. Bostanabad et al. “Computational microstructure characterization and reconstruction: Review of the state-of-the-art techniques.” In: *Progress in Materials Science* 95 (June 2018), pp. 1–41.

[27] P. Dey et al. “Informatics-aided bandgap engineering for solar materials.” In: *Computational Materials Science* 83 (Feb. 15, 2014), pp. 185–195.

[28] L. M. Ghiringhelli et al. “Big Data of Materials Science: Critical Role of the Descriptor.” In: *Physical Review Letters* 114.10 (Mar. 10, 2015).

[29] J. E. Saal et al. “Materials Design and Discovery with High-Throughput Density Functional Theory: The Open Quantum Materials Database (OQMD).” In: *JOM* 65.11 (Sept. 2013), pp. 1501–1509.

[30] E. B. Tadmor and R. E. Miller. *Modeling Materials. Continuum, Atomistic and Multiscale Techniques.* Cambridge, United Kingdom: Cambridge University Press, Nov. 24, 2011. 786 pp.

[31] P. M. Anderson, J. P. Hirth, and J. Lothe. *Theory of Dislocations.* 3rd ed. Cambridge, United Kingdom: Cambridge University Press, Jan. 16, 2017. 718 pp.

[32] J. Frenkel. “Zur Theorie der Elastizitätsgrenze und der Festigkeit kristallinischer Körper.” In: *Zeitschrift für Physik* 37.7-8 (July 1926), pp. 572–609.

[33] B. Devincre. “Three dimensional stress field expressions for straight dislocation segments.” In: *Solid State Communications* 93.11 (Mar. 1995), pp. 875–878.

[34] T. Mura. “General theory of eigenstrains.” In: *Micromechanics of defects in solids.* Springer Netherlands, 1987, pp. 1–73.

[35] J. F. Nye. “Some geometrical relations in dislocated crystals.” In: *Acta Metallurgica* 1.2 (Mar. 1953), pp. 153–162.

[36] E. Kröner. *Kontinuumstheorie der Versetzungen und Eigenspannungen.* Berlin, Heidelberg: Springer, 1958. 188 pp.

[37] E. E. Laufer and W. N. Roberts. “Dislocations and persistent slip bands in fatigued copper.” In: *The Philosophical Magazine: A Journal of Theoretical Experimental and Applied Physics* 14.127 (July 1966), pp. 65–78.

[38] J. Fish and T. Belytschko. *A First Course in Finite Elements.* John Wiley and Sons Ltd, Apr. 13, 2007. 336 pp.

[39] W. Bangerth, R. Hartmann, and G. Kanschat. “deal.II—A general-purpose object-oriented finite element library.” In: *ACM Transactions on Mathematical Software* 33.4 (Aug. 2007), 24–es.

[40] G. Alzetta et al. “The deal.II library, Version 9.0.” In: *Journal of Numerical Mathematics* 26.4 (Dec. 2018), pp. 173–183.

[41] O. Jamond et al. “Consistent formulation for the Discrete-Continuous Model: Improving complex dislocation dynamics simulations.” In: *International Journal of Plasticity* 80 (May 2016), pp. 19–37.

[42] D. W. Kelly et al. “A posteriori error analysis and adaptive processes in the finite element method: Part I—error analysis.” In: *International Journal for Numerical Methods in Engineering* 19.11 (Nov. 1983), pp. 1593–1619.

[43] L. P. Kubin et al. “Dislocation Microstructures and Plastic Flow: A 3D Simulation.” In: *Solid State Phenomena* 23-24 (Jan. 1992), pp. 455–472.

[44] M. Fivel, M. Verdier, and G. Canova. “3D simulation of a nanoindentation test at a mesoscopic scale.” In: *Materials Science and Engineering: A* 234-236 (Aug. 1997), pp. 923–926.

[45] D. Weygand et al. “Discrete dislocation modeling in three-dimensional confined volumes.” In: *Materials Science and Engineering: A* 309–310 (July 2001), pp. 420–424.

[46] G. Po and N. Ghoniem. “A variational formulation of constrained dislocation dynamics coupled with heat and vacancy diffusion.” In: *Journal of the Mechanics and Physics of Solids* 66 (May 2014), pp. 103–116.

[47] B. Devincre et al. “Modeling crystal plasticity with dislocation dynamics simulations: The ’microMegas’ code.” In: *Mechanics of Nano-Objects.* Ed. by F. Samuel, P. Anne, and T. Olivier. Paris: Presses de l’Ecole des Mines de Paris, Nov. 2011, pp. 81–100.

[48] D. Weygand et al. "Aspects of boundary-value problem solutions with three-dimensional dislocation dynamics." In: *Modelling and Simulation in Materials Science and Engineering* 10.4 (June 2002), pp. 437–468.

[49] G. Po et al. *MODEL. Mechanics of Defect Evolution Library.* URL: `https://bitbucket.org/model/model/`.

[50] W. Cai and V. V. Bulatov. "Mobility laws in dislocation dynamics simulations." In: *Materials Science and Engineering: A* 387-389 (Dec. 2004), pp. 277–281.

[51] S. Sandfeld and G. Po. "Microstructural comparison of the kinematics of discrete and continuum dislocations models." In: *Modelling and Simulation in Materials Science and Engineering* 23.8 (Oct. 2015), p. 085003.

[52] J. Rossignac. "Education-driven research in CAD." In: *Computer-Aided Design* 36.14 (Dec. 2004), pp. 1461–1469.

[53] G. James et al. *An Introduction to Statistical Learning. With Applications in R.* Ed. by G. Casella, S. Fienberg, and I. Olkin. Springer Texts in Statistics. New York, NY, USA: Springer, June 24, 2013.

[54] L. Breiman. "Random Forests." In: *Machine Learning* 45.1 (2001), pp. 5–32.

[55] A. Fisher, C. Rudin, and F. Dominici. "All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously." In: *Journal of Machine Learning Research* 20.177 (2019), pp. 1–81.

[56] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python." In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[57] D. J. Hand and K. Yu. "Idiot's Bayes: Not So Stupid after All?" In: *International Statistical Review / Revue Internationale de Statistique* 69.3 (Dec. 2001), p. 385.

[58] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning. Data Mining, Inference, and Prediction.* 2nd ed. Springer Series in Statistics. New York, NY, USA: Springer, Aug. 26, 2009.

[59] G. Ke et al. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree." In: *Advances in Neural Information Processing Systems.* Neural Information Processing Systems 2017 (Dec. 4–9, 2017). Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017, pp. 3146–3154.

[60] J. H. Friedman. "Greedy Function Approximation: A Gradient Boosting Machine." In: *The Annals of Statistics* 29.5 (Oct. 2001), pp. 1189–1232.

[61] Y. Freund and R. E. Schapire. “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting.” In: *Journal of Computer and System Sciences* 55.1 (Aug. 1997), pp. 119–139.

[62] B. Girault et al. “Strength Effects in Micropillars of a Dispersion Strengthened Superalloy.” In: *Advanced Engineering Materials* 12.5 (May 2010), pp. 385–388.

[63] D. Kiener et al. “In situ nanocompression testing of irradiated copper.” In: *Nature Materials* 10.8 (June 2011), pp. 608–613.

[64] M. Legros. “In situ mechanical TEM: Seeing and measuring under stress with electrons.” In: *Comptes Rendus Physique* 15.2-3 (Feb. 2014), pp. 224–240.

[65] D. B. Williams and C. B. Carter. *Transmission Electron Microscopy.* Springer US, 2009.

[66] Y. Zhang et al. “A method to identify dislocations in a known crystal structure by transmission electron microscopy.” In: *Journal of Applied Crystallography* 44.6 (Oct. 2011), pp. 1164–1168.

[67] A. Sumer and J. F. Smith. “A Comparison of the Elastic Constants of Chromium as Determined from Diffuse X-Ray and Ultrasonic Techniques.” In: *Journal of Applied Physics* 34.9 (Sept. 1963), pp. 2691–2694.

[68] K. Wada. *labelme: Image Polygonal Annotation with Python.* `https://github.com/wkentaro/labelme`. 2016.

[69] G. Po et al. “Recent Progress in Discrete Dislocation Dynamics and Its Applications to Micro Plasticity.” In: *JOM* 66.10 (Sept. 2014), pp. 2108–2120.

[70] H. Tang, K. W. Schwarz, and H. D. Espinosa. “Dislocation-Source Shutdown and the Plastic Behavior of Single-Crystal Micropillars.” In: *Physical Review Letters* 100.18 (May 2008).

[71] C. Motz et al. “Initial dislocation structures in 3-D discrete dislocation dynamics and their influence on microscale plasticity.” In: *Acta Materialia* 57.6 (Apr. 2009), pp. 1744–1754.

[72] C. Zhou, S. B. Biner, and R. LeSar. “Discrete dislocation dynamics simulations of plasticity at small scales.” In: *Acta Materialia* 58.5 (Mar. 2010), pp. 1565–1577.

[73] M. Stricker and D. Weygand. “Dislocation multiplication mechanisms – Glissile junctions and their role on the plastic deformation at the microscale.” In: *Acta Materialia* 99 (Oct. 15, 2015), pp. 130–139.

[74] R. Cang et al. “Improving direct physical properties prediction of heterogeneous materials from imaging data via convolutional neural network and a morphology-aware generative model.” In: *Computational Materials Science* 150 (July 2018), pp. 212–221.

[75] Z. Yang et al. “Deep learning approaches for mining structure-property linkages in high contrast composites from simulation datasets.” In: *Computational Materials Science* 151 (Aug. 2018), pp. 278–287.

[76] Dassault Systèmes. *Abaqus FEA*.

[77] S. Roy et al. “A multiscale study of the size-effect in nanoindentation of Au nanoparticles.” In: *Computational Materials Science* 162 (May 2019), pp. 47–59.

[78] W. L. Winterbottom. “Equilibrium shape of a small particle in contact with a foreign substrate.” In: *Acta Metallurgica* 15.2 (Feb. 1967), pp. 303–310.

[79] X. H. Liu and K. W. Schwarz. “Modelling of dislocations intersecting a free surface.” In: *Modelling and Simulation in Materials Science and Engineering* 13.8 (Oct. 2005), pp. 1233–1247.

[80] H. Tang, K. W. Schwarz, and H. D. Espinosa. “Dislocation escape-related size effects in single-crystal micropillars under uniaxial compression.” In: *Acta Materialia* 55.5 (Mar. 2007), pp. 1607–1616.

[81] Z. Yang et al. “Establishing structure-property localization linkages for elastic deformation of three-dimensional high contrast composites using deep learning approaches.” In: *Acta Materialia* 166 (Mar. 2019), pp. 335–345.